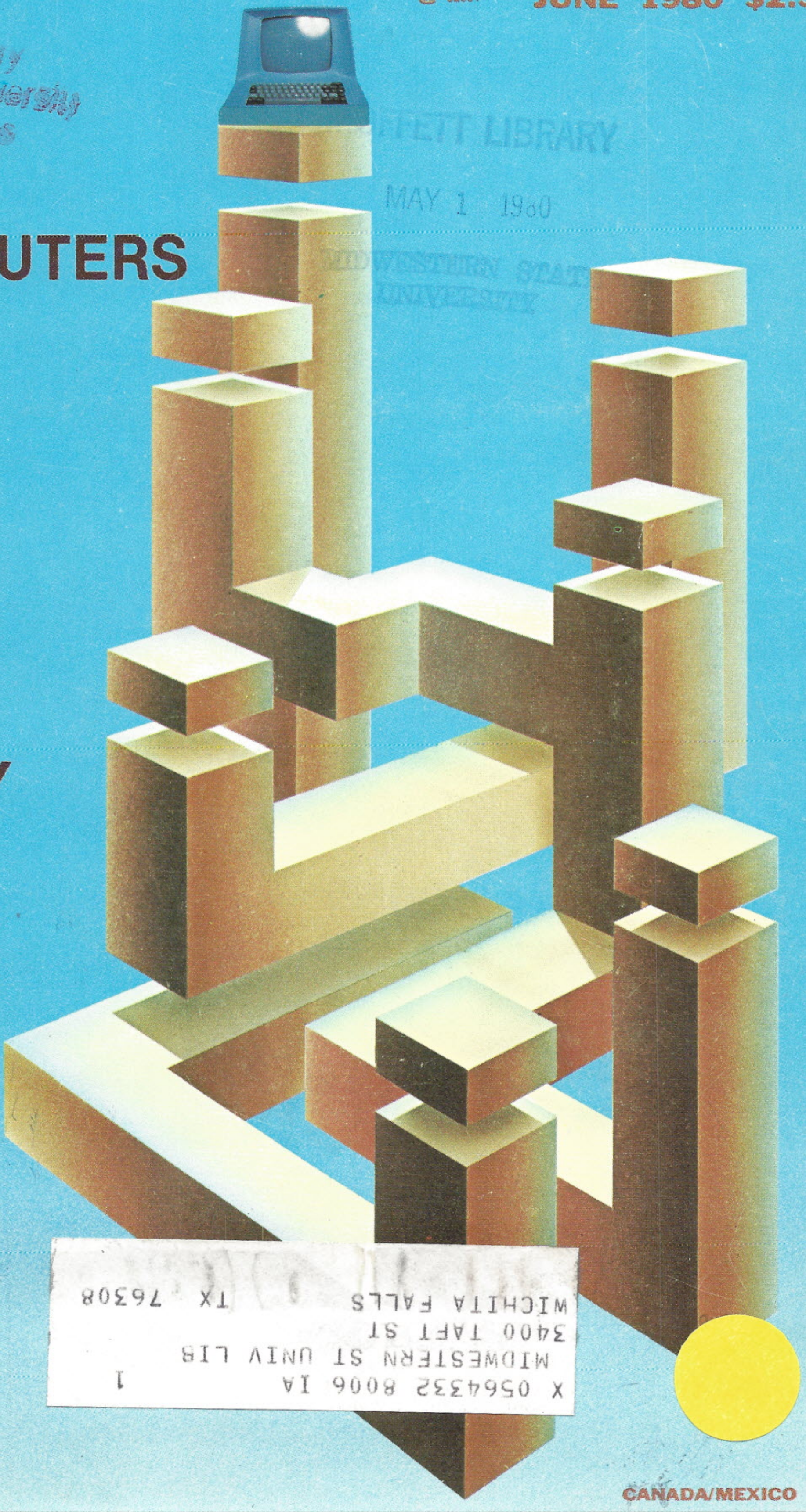
COMPUTING FOR HOME AND BUSINESS

INTERFACE AGE™

02651 JUNE 1980 $2.50

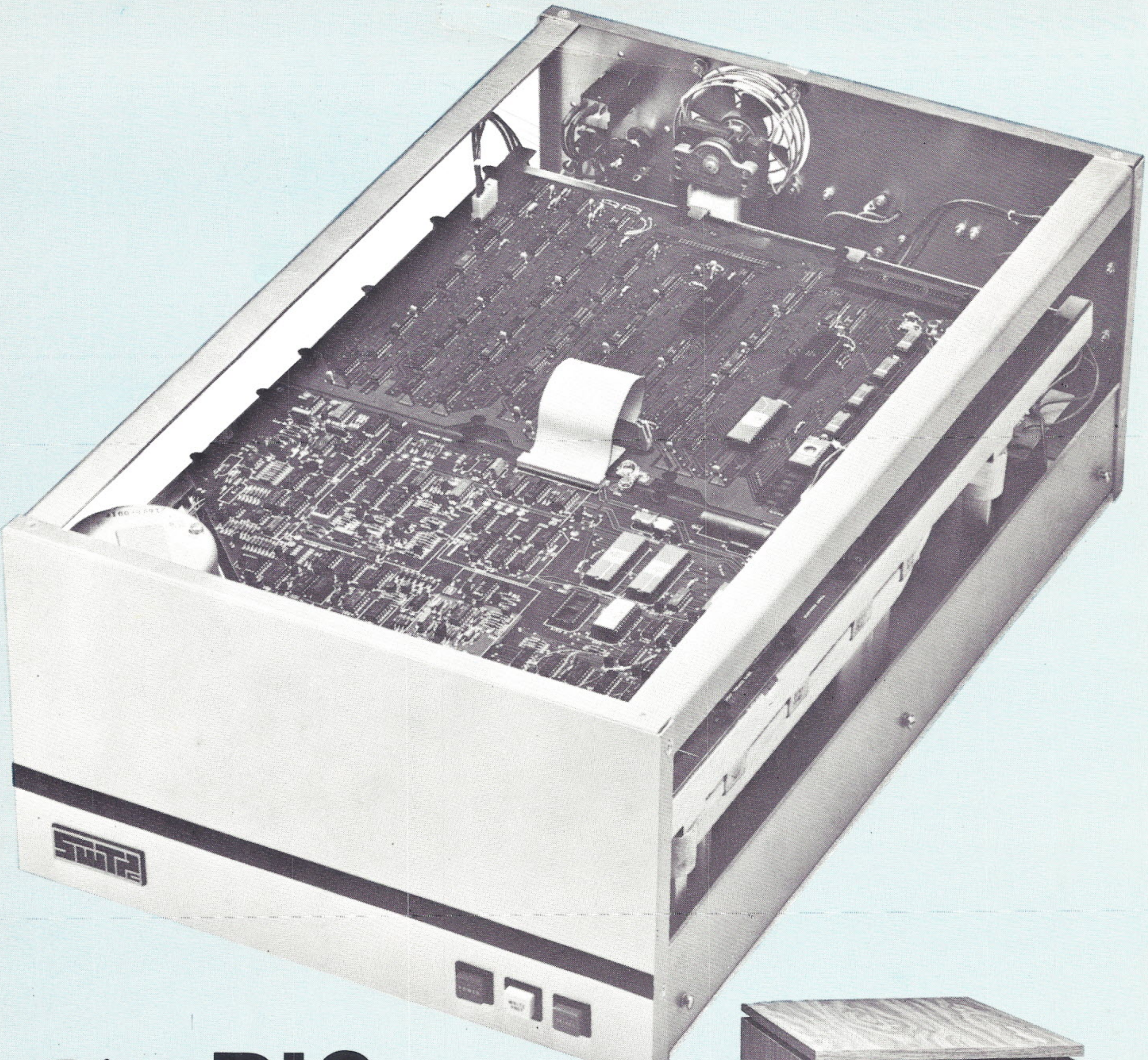


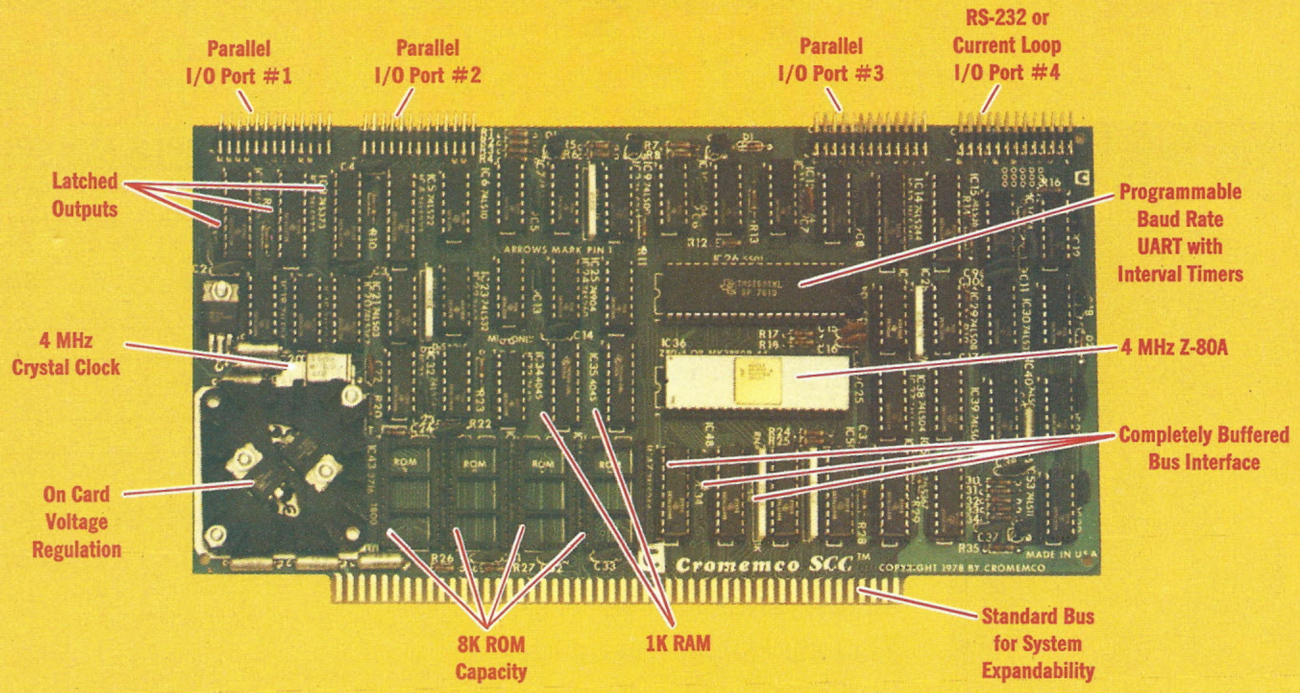
SMALL COMPUTERS OF THE 80s How Do They Stack Up?

COPS AND COMPUTERS

21st CENTURY SURGEONS

CANADA/MEXICO $3.00





The single card computer with the features that help you in real life

COMPLETE COMPUTER

In this advanced card you get a professional quality computer that meets today's engineering needs. And it's one that's complete. It lets you be up and running fast. All you need is a power supply and your ROM software.

The computer itself is super. Fast 4 MHz operation. Capacity for 8K bytes of ROM (uses 2716 PROMs which **can be programmed by our new 32K** BYTESAVER® PROM card). There's also 1K of on-board static RAM. Further, you get straightforward interfacing through an RS-232 serial interface with ultra-fast speed of up to 76,800 baud — software programmable.

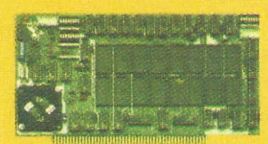
Other features include 24 bits of bidirectional parallel I/O and five onboard programmable timers.

Add to that vectored interrupts.

ENORMOUS EXPANDABILITY

Besides all these features the Cromemco single card computer gives you enormous expandability if you ever need it. And it's easy to expand. First, you can expand with the new Cromemco 32K BYTESAVER PROM card mentioned above. Then there's Cromemco's broad line of S100-bus-compatible memory and I/O interface cards. Cards with features such as relay interface, analog interface, graphics interface, opto-isolator input, and A/D and D/A conversion. RAM and ROM cards, too.

Card Cage



32K BYTESAVER PROM card

EASY TO USE

Another convenience that makes the Model SCC computer easy to use is our Z-80 monitor and 3K Control BASIC (in two ROMs). With this optional software you're ready to go. The monitor gives you 12 commands. The BASIC, with 36 commands/functions, will directly access I/O ports and memory locations — and call machine language subroutines.

Finally, to simplify things to the ultimate, we even have convenient card cages. Rugged card cages. They hold cards firmly. No jiggling out of sockets.

AVAILABLE NOW/LOW PRICE

The Model SCC is available now at a low price of only $450 burned-in and tested (32K BYTESAVER only $295).

So act today. Get this high-capability computer working for you right away.

Cromemco
incorporated
Specialists in computers and peripherals
280 BERNARDO AVE., MOUNTAIN VIEW, CA 94040 • (415) 964-7400

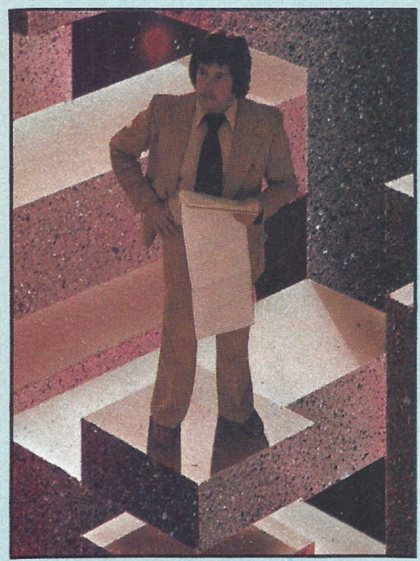
CIRCLE INQUIRY NO. 16

VOL. 5, ISSUE 6
SINCE DECEMBER 1975

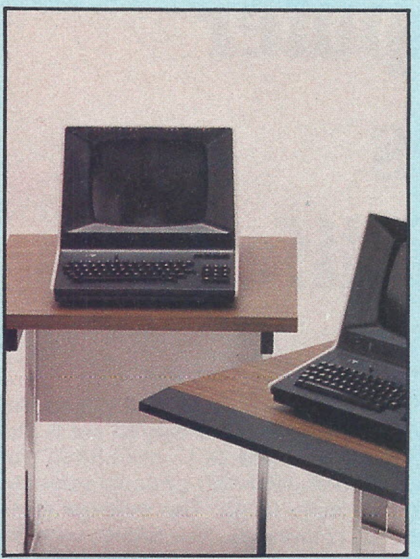
JUNE 1980

INTERFACE AGE™

COMPUTING FOR HOME AND BUSINESS APPLICATIONS



Comparison Charts . . . . Pg. 80



Looking at Micros . . . . . Pg. 70

Dust Writer . . . . Pg. 66

GENERAL FEATURES

BUSINESS FEATURES

HARDWARE FEATURES

LEARNING CENTER

SOFTWARE FEATURES

FREE RUNNING

DEPARTMENTS

INTERFACE AGE Magazine, published monthly by McPheters, Wolfe & Jones, 16704 Marquardt Ave., Cerritos, CA 90701. Subscription rates: U.S. $18.00, Canada/Mexico $20.00, all other countries $28.00. Make checks payable in U.S. funds drawn on a U.S. bank. Opinions expressed in by-lined articles do not necessarily reflect the opinion of this magazine or the publisher. Mention of products by trade name in editorial material or advertisements contained herein in no way constitutes endorsement of the product or products by this magazine or the publisher. Circulation Department, (213) 926-9540.

INTERFACE AGE Magazine is catalogued in the Library of Congress, Classification No. QA75.5.155. USPS No. 528150. ISSN Publication No. 0147-2992. Membership in Audit Bureau of Circulations applied for.
POSTMASTER: Please send change of address form 3579 and undelivered copies to **INTERFACE AGE** Magazine, 16704 Marquardt Ave., Cerritos, CA 90701. Controlled circulation postage paid at Olive Branch, Mississippi and Artesia, California.

EDITOR'S NOTEBOOK

When this issue goes to print, we will be starting at the midpoint of 1980. And where are we? As of this writing, we are facing record inflation, insane interest rates, a battle for the presidency and ever-increasing economic pressure that threatens several key segments of our economy.

One of the segments that is undoubtedly under some of the greatest pressure certainly has to be the small businessman. I'd like to address some of the challenges that are facing this very important part of our nation's foundation.

As we see the economic noose tightening on the small businessman, several things are going to begin happening—and it will not require a soothsayer to predict the occurrences. Undoubtedly, we will see more and more small businesses failing. After all, some 70 to 80% of all new small businesses fail in their first five years even when the economy is relatively stable. Surely the small business failure rate will continue to climb.

We will probably see fewer new stores, shops and businesses sprout into being, as the cost of seed money is entirely unreasonable. One of the realities of life is that it generally takes money to make money, and if the budding entrepreneur has to borrow money at prohibitive interest rates, then the cost of money alone will probably sound the death bell over new business.

But don't get me wrong, I'm not mourning the demise of the small businessman. On the contrary—the typical small businessman or entrepreneur is no slouch. These are people who are dedicated to the success of their venture, people who are not afraid to take a risk, people who will be more than willing to turn to technology for assistance in meeting the challenges of small business.

Many of these people will turn to the micro-based small business system, hoping to use the "electronic brain" of the 1980s to their economic advantage.

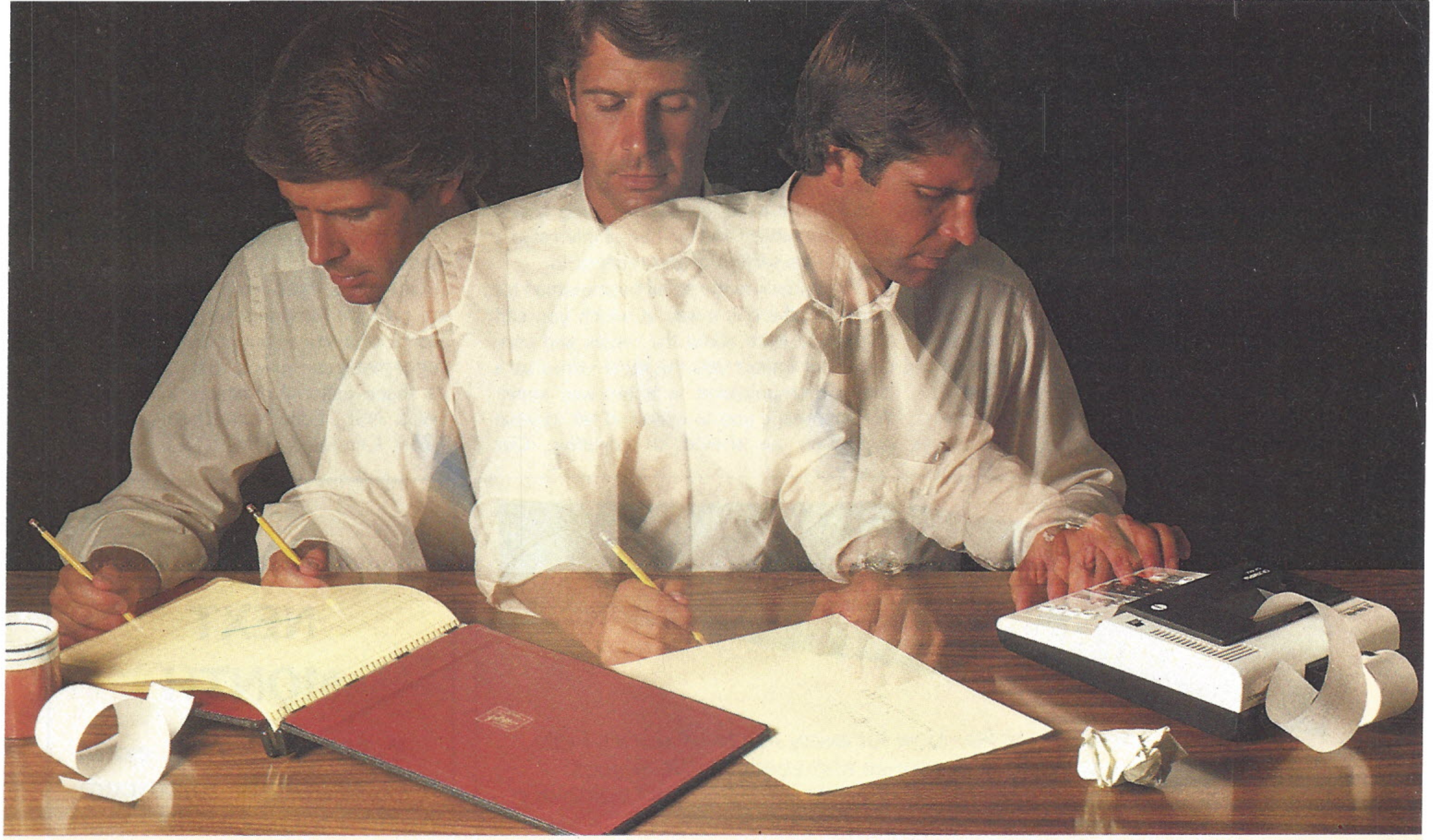
A number of small businesses will succeed in the marriage of small business and the micro. After all, one of the reasons for the booming success of the small system is that it is an extremely powerful tool for small business and the cost is not necessarily prohibitive.

But what of those businessmen who are unsuccessful in their implementation of the micro-based system? Does their failure to implant high technology within their business reflect only an individual failure, or is it a failure of our industry to provide a "universal" enough machine for every business, every situation, every application and every individual? I suspect that it is neither, for there are few simplistic answers for multi-faceted problems.

One of the causes that we will be examining in future issues has to do with planning. The fact of the matter is that many of the planning activities that are essential for the successful operation of the largest computer centers are just as necessary for the small businessman. In fact, some of the planning is even more critical for small businesses because of the general lack of a broad financial base within individual businesses.

For despite wishes to the contrary, the installation and implementation of a computer system within the small business framework is not a "plug and go" situation. While technology of 1980 can be a Godsend to the small business, it can also be a disaster of the gravest proportions if a system is ill-conceived and poorly planned.

Specifically, we are going to start examining the roles of education, backup, disaster planning, pre-purchase evaluations, systems analysis, ethics, privacy and many other issues. We will be looking at how these issues can and do impact small businesses and how the users of micro-based systems can benefit greatly from the trials and errors that large systems users have been dealing with for many years now.



The additional attention that we will be giving to the use of the micro in small business does not represent a departure from the present editorial direction of INTERFACE AGE, instead, we view it as an extension of the existing direction. We have no intention of ignoring the needs of those readers who are not involved in small businesses. We intend to continue our coverage of all the diverse aspects that make microcomputer use interesting, challenging and fun.

I encourage those readers whose primary interest is in the personal computing arena to pay some heed to this upcoming coverage. This subject area will be of interest to a broad cross-section of our readership.

Getting back to the small businessman for a second, there is a way in which you can help us to write about the issues and concerns that impact you the most—drop us a line. Communication is a two-way street. We encourage you to write and tell us what you think we should be covering. Our publication is only as good as our readers, and we are keenly interested in writing about the subjects that interest you the most, not just the subjects that happen to tickle our fancies.

Of course, there is no guarantee that we are going to jump at a suggested subject in search of a story—but your comments and suggestions can play a vital role in helping us shape editorial direction. Write INTERFACE AGE, Editor's Notebook, P.O. Box 1234, Cerritos, CA 90701.□

COMING NEXT MONTH

INTERFACE AGE examines a unique microcomputer application that monitors and controls a household solar collector. In addition, those with conventional heating devices will learn how to apply this system to their individual needs.

Remote control is discussed as a way to give a system maximum control over home or business functions.

A new series on computers in education will be introduced as INTERFACE AGE explores the expanding role of computers in the classroom.

The July issue also features the monthly software, business and hardware stories, columns and tutorials. But that isn't all; a special new products directory is offered, listing over 250 products, making this a valuable and highly informative issue.

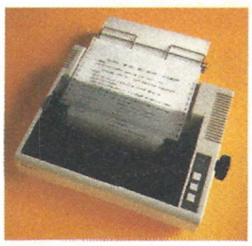
One company has sold more printers to this planet than anybody.

Epson.

Surprised? You shouldn't be. Because we've manufactured over half the print mechanisms in existence on this planet. And we've sold more of them than all the others—combined. Now that's the kind of experience you can count on.

But maybe you haven't heard of us. And that's understandable. You see, for years, different companies have marketed the lion's share of our print mechanisms for hundreds of applications in dozens of markets. Our products, their names. Now we're changing that: our product, our name.

We'll tell you right now that we intend to be as big in printers for personal computers as we are in printers for the rest of the world. And we intend to get there the same way. By making printers you can count on to perform. With the options and interfaces you need. And by delivering what we promise. When we promise it.



The funny thing is that we never set out to be the biggest printer company in the world—only the best. And we didn't get here by turning out inferior products or charging an arm and a leg. We got here by turning out quality mechanisms at the rate of one for every second of every working day. So we can sell each one for a little less.

You'll find Epson in better computer stores everywhere. Look for us. Because even if you haven't heard of us up to now, you'll be hearing a lot more of us from now on.

EPSON
EPSON AMERICA, INC.

Western: 23844 Hawthorne Boulevard, Torrance, California 90505 • (213) 378-2220 TWX: 910-344-7390
Eastern: 98 Cutter Mill Road, Rm. 350, Great Neck, New York 11021 • (516) 487-0660

INTERFACE AGE

16704 Marquardt, Cerritos, CA 90701
(213) 926-9544
TWX (910) 583-1412

PUBLISHERS
PUBLISHER/EDITOR-IN-CHIEF **ROBERT S. JONES**
EXECUTIVE PUBLISHER **NANCY A. JONES**

ADMINISTRATION
PUBLICATION DIRECTOR **MIKE ANTICH**
PUBLICATION ASSISTANTS **DORIS RIOPEL**, **CHERYL JOHNSTON**
ACCOUNTING SUPERVISOR **KAY SOTO**
ACCOUNTING ASSISTANTS **MARY ANN LOWER**, **SHIRLEY MAZENKO**

CIRCULATION
CIRCULATION MANAGER **ED UECKER**
CIRCULATION SECRETARY **JEANNE ADDINGTON**
CIRCULATION ASSISTANT **CHARLOTTE SEVEDGE**

EDITORIAL
EDITOR **MICHAEL PANCHAK**
ASSISTANT EDITORS **KATHY TEKAWA**, **LES SPINDLE**
SENIOR EDITOR SOUTHEAST REGION **BILL TURNER**
SYSTEMS EDITOR **TOM FOX**
SOFTWARE EDITOR **ALAN R. MILLER, PhD**
ASSOCIATE EDTIORS **HENRY DAVIS, AL BAKER**
CONTRIBUTING EDITORS **R. W. BEMER**, **ROGER C. GARRETT**

Editorial Correspondence
Direct all correspondence to the appropriate editor at: INTERFACE AGE Magazine, P.O. Box 1234, Cerritos, CA 90701.

PRODUCTION
PRODUCTION MANAGER **TERRI LEDESMA**
ART DIRECTOR **FINO ORTIZ**
ARTIST **SAMANTHA LEE**
TYPOGRAPHER **MELODY A. MARTENS**

ADVERTISING
NEW ENGLAND REGION **DICK GREEN**
7 Lincoln St., Wakefield, MA 01880 (617) 245-9105

EASTERN REGION **JOHN SENSENSTEIN**
20 Community Pl., Ste. 140, Morristown, NJ 07960
(201) 267-3032

SOUTHEAST REGION **HARRY DILL**
3938 Sussex Avenue, Charlotte, NC 28210 (704) 552-1004

MIDWEST REGION **AL GRAVENHORST**, **STEVE SKINNER**
5901 N. Cicero Ave., Chicago, IL 60646 (312) 545-8621

WESTERN REGION **BRUCE BERKEY**, **ZACH BOVINETTE**
P.O. Box 4566, Pasadena, CA 91106 (213) 795-7002

JAPAN **TOMOYUKI INATSUKI**
Trade Media Japan Inc., R. 212 Azabu Hts., 1-5-10, Roppongi, Minato-ku, Tokyo 106
Telephone: (03) 585-0581 Telex: J28208

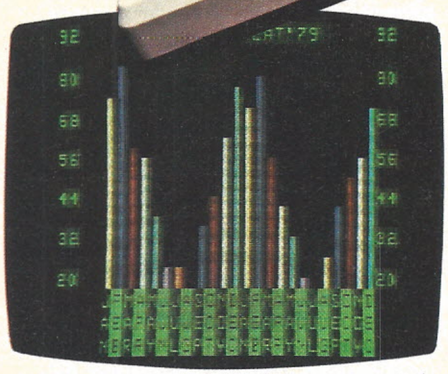
INTERFACE AGE EUROPE
DIRECTOR, EUROPEAN OPERATIONS **H.L. GROHMANN**
Dahlienstr. 4, D-8011 Munchen-Vaterstetten
West Germany
Telephone: 08106/7396

INTERNATIONAL NEWSSTAND DISTRIBUTION
LEW ULLIAN
Orberstrasse 38, D-6000 Frankfurt/M. 61, West Germany
Telephone: (0611) 44 77 90/41 84 80

INTERNATIONAL/DOMESTIC RETAIL CIRCULATION
MARY ANN LOWER (213) 926-9544

MEMBER OF THE WESTERN PUBLICATIONS ASSOCIATION wpa

AMERICAN SOCIETY OF BUSINESS PRESS EDITORS MEMBER



The only computer with color, sound, user programmability and expandability at $599.

The Imagination Machine offers more at its price than any other personal computer on the market today.

Consider these features: 9K RAM, with 14K BASIC in ROM,53-key typewriter keyboard. A fine resolution picture, generated on your television set or monitor in 8 colors!

A built-in, dual-track cassette tape deck with 1500 baud rate, for APF's digitally recorded, "saturated," tape programs. A built-in sound synthesizer. And two, built-in, game style controllers, with joysticks and numeric keypads.

When you want to go beyond APF's library of educational, home-and-personal management or entertainment programs... when you want to create your own programs... you can. The Imagination Machine is programmable in BASIC and 6800 machine language. The Imagination Machine is also expandable. Just add our "Building Block", an optional, four-port expansion device, and you can hook up a printer, telephone modem, and additional memory cartridge or mini-floppy disk drive.

For the name of your nearest Imagination Machine dealer call, TOLL FREE: 1-800-223-1264. (New York residents call: (212) 758-7550) or write: APF Electronics, Inc. 444 Madison Avenue, N.Y., N.Y. 10022.

$599. Manufacturer's suggested retail price.

APF electronics inc.

CIRCLE INQUIRY NO. 2

See us at NCC Booth 85

"YOUR LIFE WILL NEVER BE THE SAME!"

LETTERS

PEEVED BUYER

I would be interested in reading an article in your magazine telling the businessman how to obtain a microcomputer. I'm not speaking of applications of various types of software; I'm speaking about just the simple acquisition of one.

I have been trying to acquire a system for use in my business since last July. On five occasions, either I or my secretary have gone to Radio Shack or Apple stores by appointment to observe a demonstration. The sales people twice failed to be in the store. At no time were we ever given a demonstration. One Apple salesperson did show me a beautiful printed circuit board. I truly fail to understand why I am having such a difficult time spending a minimum of $5,000.

It is obvious to me why IBM is the leader in its field; only IBM has bothered to phone me numerous times and tried to sell me a $22,000 system. When IBM produces a unit for less than $10,000 we will witness the immediate death of all other microcomputer manufacturers, especially Radio Shack and Apple.

David Dominick
Orange, CA

CPU PROBLEMS CITED

Regarding your January Inventor's Sketchpad by Roger Garrett, multiprocessing seems to be the way of the future, but Mr. Garrett soft-pedals the real problems of multiprocessing:

1. Complexity of operating system and monitor software are increased. In conflict are desires for shared resources to be fully used, and that contention be avoided. The question becomes one of serialization of activity vs. courting either "deadly embrace" lockups or a lack of integrity.
2. The "nearly double" power from two CPU-multiprocessing is substantially worse than that, and returns diminish for each CPU added.
3. Reliability decreases; simply, there are more interdependent parts to fail. Should one CPU fail holding locks for serialization, the other processors face eventual lockout, unless some very delicate recovery is performed. Additionally, physical and electronic interdependencies may cause one CPU's ills to become system ills, or for the whole machine to be taken for maintenance.

Darrell Jones
Eugene, OR

ON NEW PROM APPLICATIONS

The article, 'Powered Down Bipolar PROMs' in your April issue struck a resonant chord, so I did a little research and found a similar idea in National's Memory Data Book-1977. The Data Book should be consulted for additional information on this very useful concept. The information was for 74S287 PROMs, but I plan on incorporating the idea using 74S472s (512 x 8 in a 20 pin package); they make excellent patches for the MCM66714 character generator.

For those who hate discrete components, the SN75327 by Texas Instruments comes in a 16 pin DIP package, contains four independent switches, each capable of sourcing up to 600 mA and provides switching times of 30 ns as well as a common strobe input. A dual PNP driver will fit on a 16 pin header plug, whereas the 75327 provides twice as many switches. Two may be used, if preceded with inverters, as direct replacements for the eight Power Switchers needed to program the 93448.

Many thanks for the hardware article; it saves me almost 2 watts of power.

Gerald R. Pomraning
Wilder, ID

CROMEMCO'S SUPERDAZZLER

Thank you for Tom Fox's informative and exciting article on Cromemco's Superdazzler. I have some uncertainties regarding memory access and allocation:

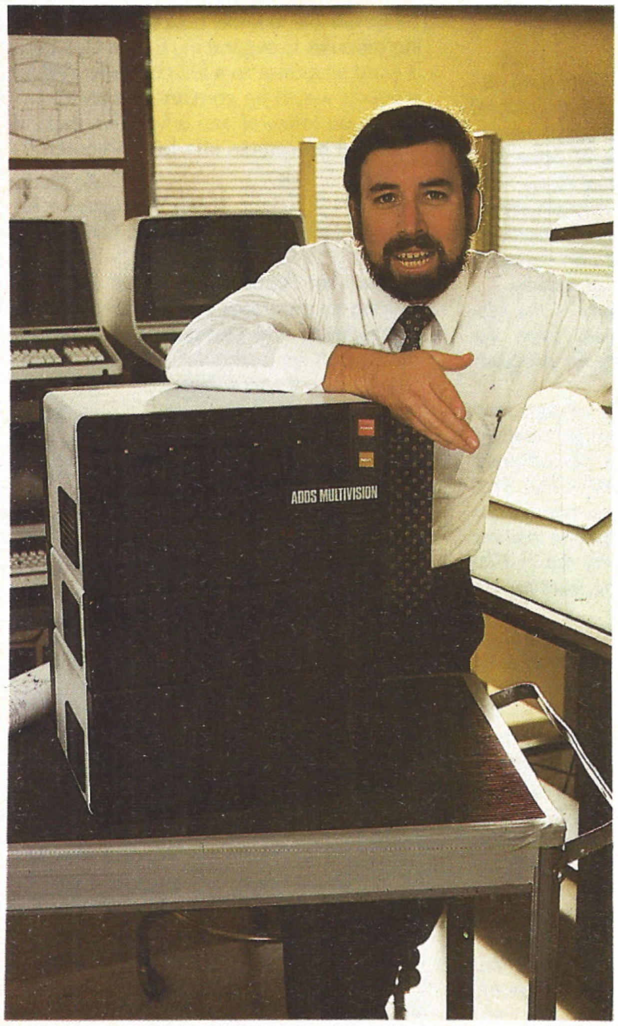
1. Is it correct to say that the Superdazzler cannot operate without some form of direct memory access?
2. If the Superdazzler is hooked up with DMA, but without two-port RAMs, and if there is plenty of computer memory (e.g., more than 64K), is it correct to say that the main deficiency is the relative overload of the S-100 bus so that the program complexity is reduced and execution time is increased?
3. If the Superdazzler is hooked up with DMA and with 48K of 2-port RAM memory, does it still require 48K of RAM picture memory in the computer (in addition to at least 12K of program memory), or does the computer memory now require only program memory (e.g. 12K or more)?
4. Why does the DMA board have to be connected with computer memory at all? I assume this connection is needed for program operation. But, could the program memory in the computer be connected to the SDI via its video board? Then, the DMA could be connected to its own picture memory RAM without interference with S-100 bus or computer memory. I assume, however, that the "picture memory" RAM might require input from an I/O port. Could this be done via a 2-port picture memory RAM, again leaving the computer memory free for programming?

I suppose my distinction between picture memory and program memory is somewhat artificial, but it would be instructive for me to know where it breaks down.

5. What is Cromemco's memory mapping scheme allowing several banks of 64K memory to be contained in the computer? Is this related to Cromemco's 64K

"INTRODUCING THE HOTTEST THING OFF THE DRAWING BOARDS: A COMPUTER TRIO THAT TAKES YOU FROM SMALL TO BIG."

Geof Karlin
Director of Systems Development



If you're considering your first computing system, you need to know more than what it will do for you today. You need to know what it will do tomorrow.

Many businesses outgrow their first computer within a year or two. And when they do, they find their system is difficult if not impossible to add onto–because adding on requires a different manufacturer's equipment, different operating system, different programming languages.

At ADDS, we've just solved this problem with Multivision, a compact trio of stacking computers with ADDS' CP/M®–compatible, multi-user operating system. You can begin with one and expand as needed.

MULTIVISION 1 (top module) is a get-started computer with 5 MHz processor, 64K bytes of RAM (Random Access Memory) and floppy-disk storage capacity of 700K bytes. It's available with a wide range of business application software. We even offer our own word processing package. A fully loaded Multivision 1 is list-priced at $3,785 without terminal.

MULTIVISION 2 (top and bottom modules) uses the new Winchester technology to provide 5M or 10M additional bytes of hard-disk storage. List-priced at $7,995 for 5M bytes of disk, it is thousands less than other hard-disk systems.

MULTIVISION 3 (entire stack) turns your computer into a multi-user system with up to 256K bytes of RAM that supports up to four display terminals. It allows four different parts of your business–i.e., accounting, marketing, purchasing, and shipping–to share data and simultaneously use the system. No other microsystem offers so much for so little.

Before you decide upon any small computer, look into ADDS Multivision. For years we've been the largest supplier of display terminals to computer giants.

Now we're making a system for *you*.

For information, write: Systems Division, Applied Digital Data Systems Inc., 100 Marcus Boulevard, Hauppauge, N.Y. 11787. Dealer inquiries invited.

CP/M is a registered trademark of Digital Research, Inc.

ADDS

ADDS
Applied Digital Data Systems Inc.
SOMETHING EXTRA IN EVERYTHING WE DO

MULTIVISION

RAM card with extended bank select (Model 64 KZ)? If so, what is bank selection?

I would appreciate any advice or references you can give me.

John Silver
New York, NY

1. *Although Cromemco's SDI circuits utilize the DMA technique in their internal operation, there is no particular necessity that the host computer have any special talent for this sort of thing.*
2. *You are correct — the only disadvantage is that of speed. A major advantage of not purchasing the two-port memory is that you will save a lot of money.*
3. *No — the two-port memory is the picture memory and, if filled, frees the normal computer memory for program and/or data retention.*
4. *Here, I believe, is the crux of your misunderstanding. The picture (as retained in the picture memory) is placed there by a program running in the CPU. Program memory is, of course, needed for any such program to run. The picture memory sets its picture via the S-100 bus, and transmits it to the monitor via the SDI two-card set.*
5. *The memory mapping scheme is a popular method for fitting more than 64K of memory to a CPU which would otherwise be limited by its 16-bit addressing bus to that amount only. Each memory card fitted with the bank select feature can be turned on and off by signals from the CPU. Since more than one memory card can be addressed at the same location within the 64K address space, the CPU is careful to only turn one of them on at any given instant. It's a handy (but, again, expensive) way to keep a picture on ice and yet instantly available for display on the monitor. Memory mapping/bank select has wide usage in multi-terminal microcomputer systems.*

Tom Fox

DP: QUESTIONS & ANSWERS

I am writing for information on microcomputer controls such as those featured in the microwave oven, electronic toys, digital controlled television sets, and other appliances. I have several inventions and would like to incorporate the digitial control technique into them.

I have no knowledge of electronics, but some knowledge of programming and system operations. I can't find a school or institution that teaches digital controls using the microcomputer. I would appreciate any information on the teaching or building of single chip microcomputers used specifically for digital controls.

Don D. Wilson
9055 S. Luella
Chicago, IL 60617

Do you know of any software and/or hardware designed for academic and/or admissions offices of independent schools?

Louis A. Young III
P.O. Box 657
Pebble Beach, CA 93953

I would like to know how to go about adding modules to expand my Exxon Qyk Level 3 word processor to a full computer. Which products would be compatible with it?

Do you know of any articles on expanding word processors into full computers?

C. A. Bodor
2699 Youngstown Rd.
Warren, OH 44484

I own an Apple II computer and I am looking for programs on I Ching, astrology, numerology, and biorhythms. I also need information on computer portraiture methods, hardware and software.

Thomas A. Waye
2525 NW 105 Lane
Sunrise, FL 33322

We have published full addresses to these letters so that our readers may offer assistance. For books on these subjects write Data Dynamics Technology, P.O. Box 1217, Cerritos, CA 90701, for their catalog.

WHERE CREDIT IS DUE

All the photos in the article Micro Mixdown — There's a 65K Helper in the Recording Studio (INTERFACE AGE, January 1980) are of the AUTOMATT in San Francisco, which was not stated in the article.

Photo 1 is Studio C, where a Harrison 40-in/32-out console is interfaced with an Allison 65K Programmer. The Allison is in turn interfaced with a Zilog Z-80 microcomputer, which is used to display the status of the various (Harrison) console functions and/or the corresponding data for those functions on the multi-track recording tape.

In addition, data and text (artist's names, channel assignments, etc.) related to a particular song or "cut" may be displayed instantly on the console.

Studio A and B also feature automation, and Chief Engineer Michael Larner is the man who put it all together.

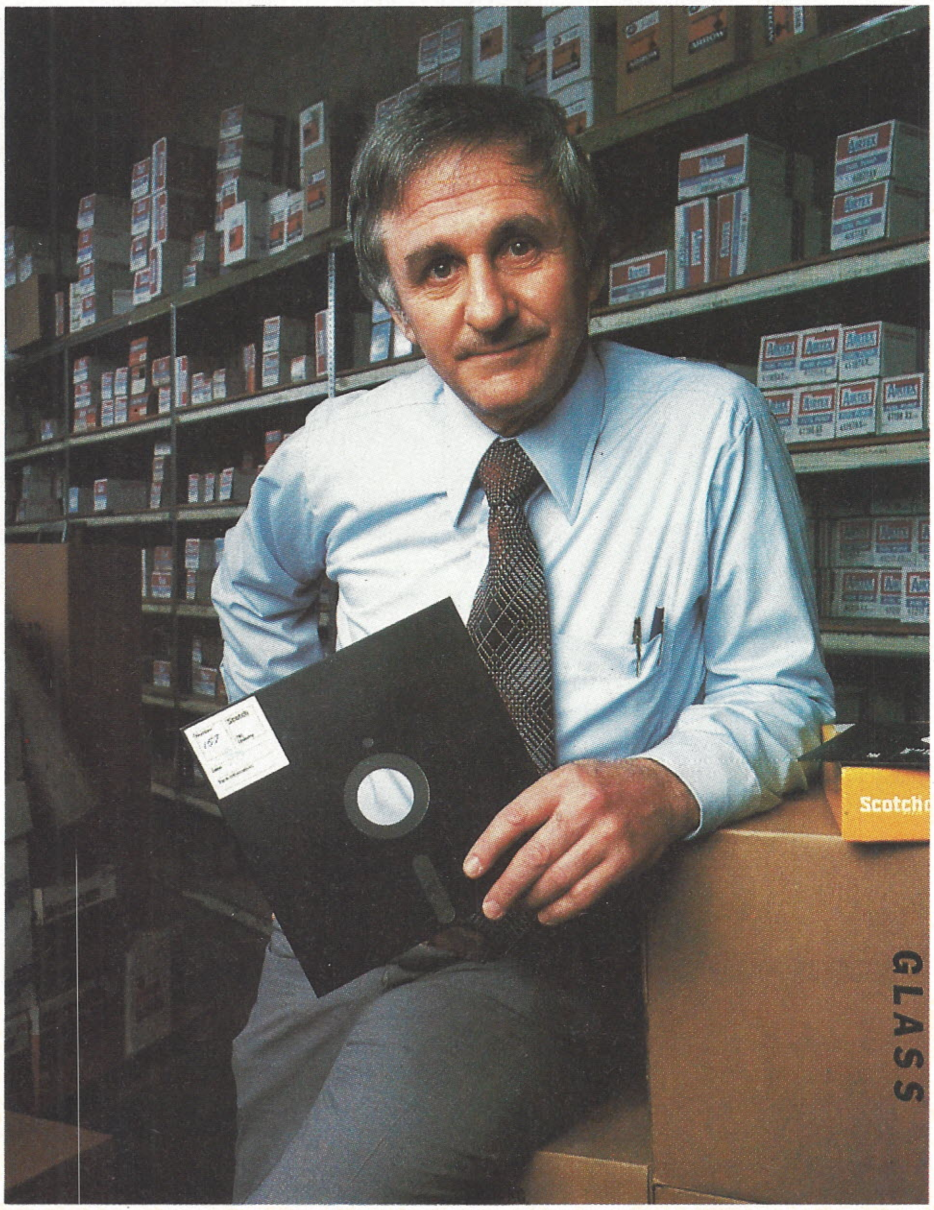
Steve L. Martin
Santa Cruz, CA

DEFINING 'INTELLIGENCE'

In Roger Garrett's April article, "A Discussion of Artificial Intelligence — A Definitive Answer to the Question: Can Computers Think?", his approach is interesting. The scope is the whole of human knowledge and experience. I would like to offer the following observations to extend Mr. Garrett's definition.

Consider a system with internal and external compartments and a transducer at the interface of the two regions. Output from the

"Our inventory is our existence. Think we'd trust it to anything less than Scotch® Brand Diskettes?"



Don Stone, President, Mass. Auto Supply Company, Inc., Boston, Mass.

Scotch Diskettes are the diskettes you can depend upon with the information *your* business depends upon.

Each one is tested and certified error-free before it leaves our factory. Because we know nothing less than perfection is acceptable for *your* vital business data.

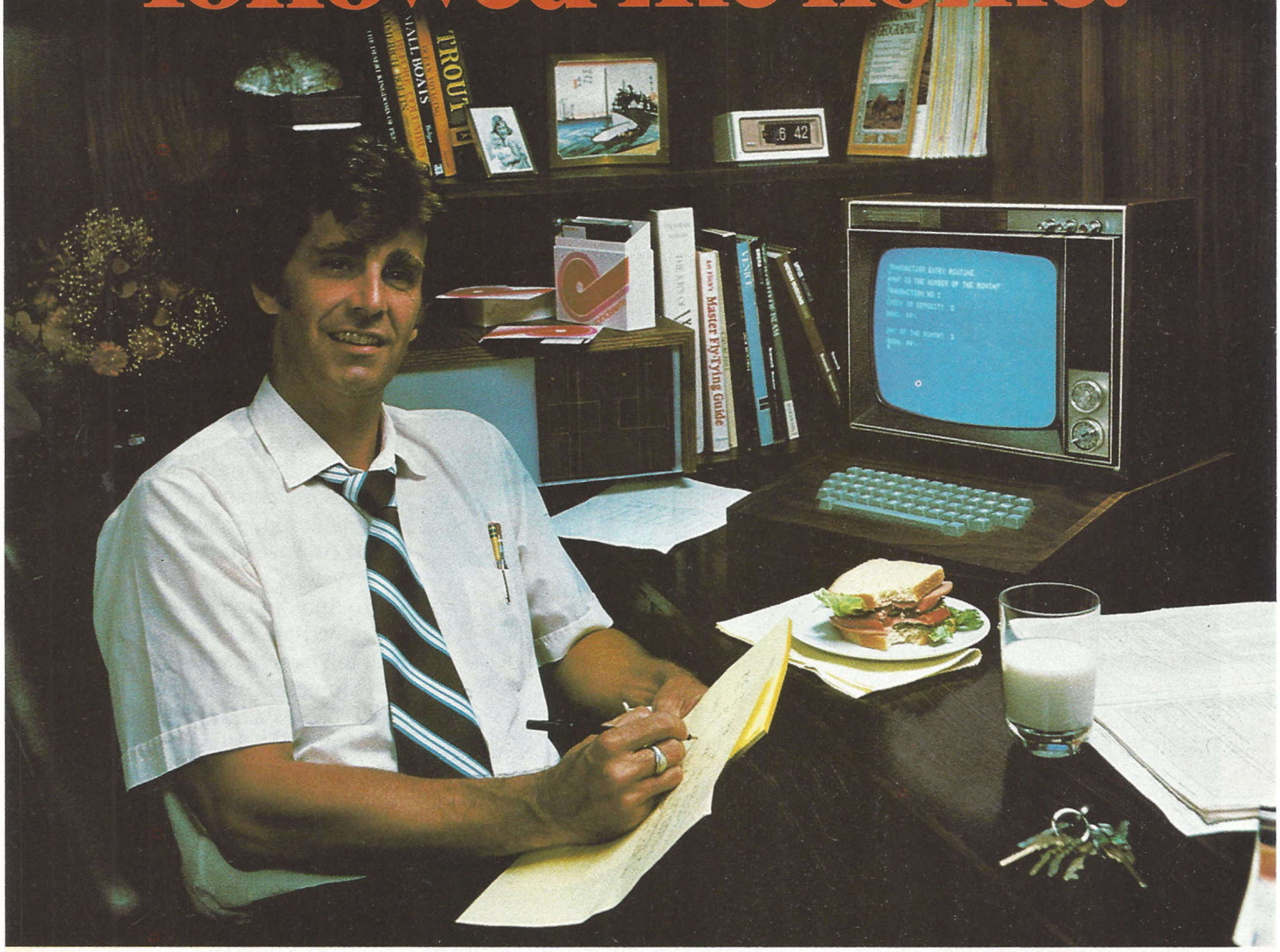
Scotch Diskettes are available in regular or mini sizes, compatible with almost any system.

To find out where you can purchase Scotch Diskettes, call toll free: 800-328-1300. (In Minnesota, call collect: 612-736-9625.) Ask for the Data Recording Products Division. In Canada, write 3M Canada Inc., London, Ontario, N6A 4T1.

If it's worth remembering, it's worth Scotch Data Recording Products.

3M

"My Shugart followed me home."



"After working all day with the computer at work, it's a kick to get down to Basic at home. And one thing that makes it more fun is my Shugart minifloppy™. We use Shugart drives at work, so when I bought my own system I made sure it had a minifloppy drive.

"Why? Shugart invented the minifloppy. The guys who designed our system at work tell me that Shugart is the leader in floppy design and has more drives in use than any other manufacturer. If Shugart drives are reliable enough for hard-working business computers, they've got to be a good value for my home system.

"When I'm working on my programs late at night, I can't wait for cassette storage. My minifloppy gives me fast random access and data transfer. The little minidiskettes™ store plenty of data and file easily too.

"I made the right decision when I bought a system with the minifloppy. When you lay out your own hard-earned cash, you want reliability and performance. Do what I did. Get a system with the minifloppy."

If it isn't Shugart, it isn't minifloppy.

Shugart

435 Oakmead Parkway, Sunnyvale, California 94086

See opposite page for list of manufacturers featuring Shugart's minifloppy in their systems.

transducer is received by a network composed of various levels of functional units that receive input from the transducer or other levels in the network with a final output pattern resulting. In a simple representation, this could be a photocell to turn on the porch lights at dusk, or a human finger's temperature receptor responding to hot water and a verbal expression of "ouch."

If we consider the ultimate in information about an external event to be the information obtainable from the quanta of energy given off, we immediately are confronted with the limitations of information gathering. Entropy, Heisenberg's uncertainty principle, the physical nature of the transducer, and various other factors place a limit on how much is directly knowable about any event. Thus, the transducer can be considered to carry out a kind of mapping function. A one-to-one mapping function associates every element in one group with a single element in a second group. In the case of a transducer, we would need an efficiency conversion of 100%.

There is a loss of information, so the transducer is of a many-to-one class in its action as a mapping function. That is, some of the information input in the form of energy quanta are indistinguishable and would be mapped onto the same output signal.

The levels of the human nervous system can also be considered to carry out mapping functions. They, too, are of a many-to-one type. Thus, there is a loss of information regarding an external event before we begin processing at the level of human thought.

I would like to simply state that human language is not always a one-to-one mapping, either in its 'internal' use or when used to communicate information to others.

Thus, the idea of restricting a hypothetical thinking machine to human language processing seems a severe handicap for the machine. Trying to define a new, more precise language or to develop a programming language to solve this problem is approaching the problem in reverse. An intelligent machine should get information from the source.

My use of the concept of mapping is similar to Mr. Garrett's definition of thinking. Equating thinking to mappings of information to another pattern allows some things to be defined as thinking which are not usually considered as such. There is a wide separation, quantitatively if not qualitatively, between simple machines and humans. There is another qualitative aspect to the definition of thinking which creates a new category, that is the ability to generate and implement a new mapping function (in whatever form — 'hardware' or 'software').

I commend Mr. Garrett on his informative and stimulating article, and on his positive approach to the subject.

Darwin Kenepp, M.D.
Penn-Wynne, PA

Your observation of the tendency to equate language processing with intelligence is well founded. Early attempts involved simple syntactic analysis, the derivation of denotation from text. Semantic analysis, the derivation of connotation, was also required in order to determine meaning rather than simple structure. When this failed, other methods were employed. One method describes "scripts," or simple environments, within which the analysis takes place. This restricts the computer's world view, so that it knows what to expect from the text.

The problem with all of these methods is that they are approaching the concept of intelligence from too high a level. Simulating the expressions of intelligence is not the same as simulating intelligence itself. It is the phenomenon of intelligence, or thinking, which I was attempting to define in my article and which you have described utilizing a slightly different approach.

You have identified perhaps the greatest obstacle in the acceptance of this approach when you observe that this view "allows some things to be defined as thinking which are not usually considered as such." The obstacle stems from the distaste most people have for a mechanistic/deterministic view of the world and the presumed "special place" that man has in some ethereal "scheme of things."

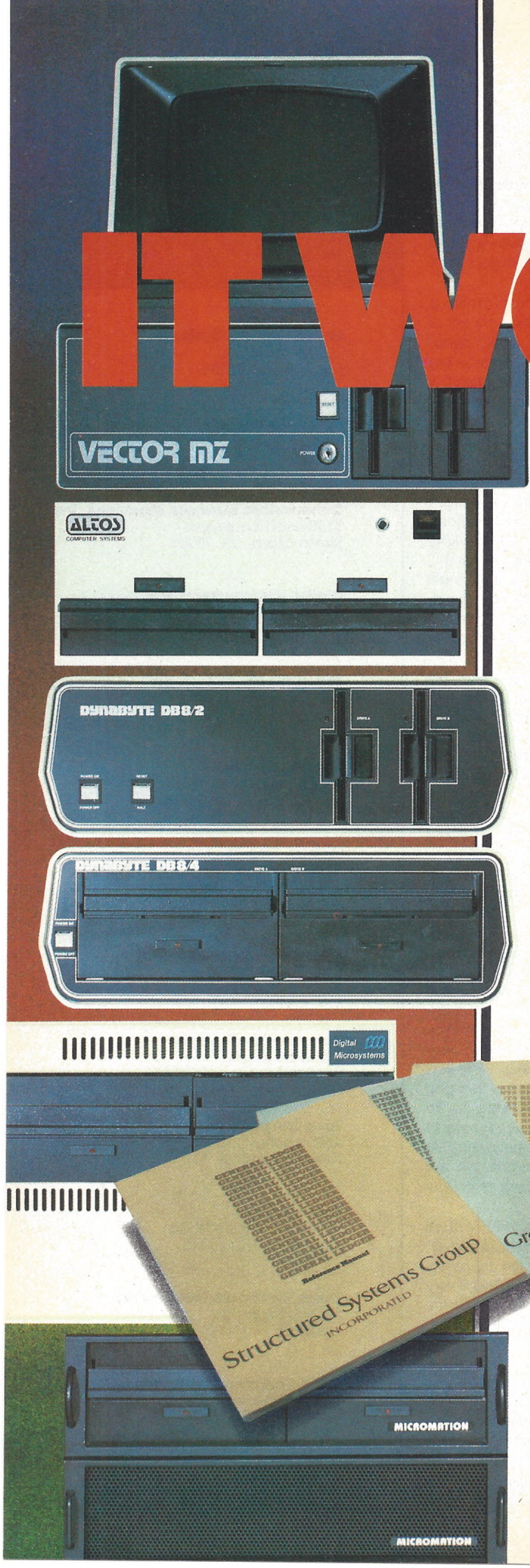
The assumptions used here are (1) human beings are special, and (2) the ability to think, to emote, to create original ideas is that which makes us special. The conclusion is: non-human entities cannot possibly think, because it would contradict the assumptions.

In approaching the issue from a scientific perspective, however, we can logically make no such assumptions. We attempt to define consistencies, to observe what exists, to measure and count, and to draw logical conclusions regardless of preconceived notions.

Our conclusions are often expressed as models of reality, hopefully a model which comes close to that reality. Your mapping model and my own data/attribute/relationship model are two approaches. Perhaps one of these is close to reality; we cannot tell how close. The results may very well allow some things to be defined as thinking which are not normally considered to be thinking. But, when we are truly aiming to reveal the truth, not merely attempting to substantiate preconceived notions, we are willing to accept the consequences of that understanding.

I see your model as a description of mine at a much higher level of conceptualization. Language processing models (an extremely high level of thinking) have not worked very well. My model, purposely defined to be the lowest possible level of conceptualization, may prove to be difficult to implement with current technology. Your model is somewhere in between these extremes, and may prove to be more easily implemented to include expressions of intelligence such as emotion, creativity, self-consciousness and the desire to search for truth.

Roger Garrett



The presidents of the companies making these computers recommend Structured Systems Business Software for one very simple reason:

IT WORKS.

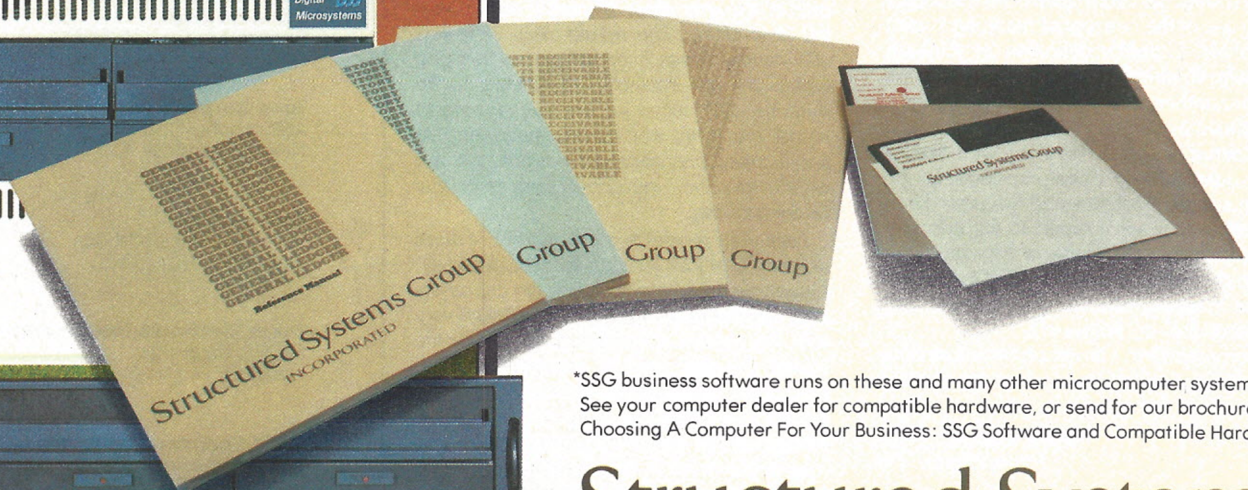
Structured Systems Business Software works for business. It works as delivered, without customization or reprogramming. It works without extensive retraining, or any new data processing personnel.

And it works hard. Our business software turns a computer into an efficient, hardworking employee, to help your business work better.

We've got working solutions for your General Ledger, Payroll, Accounts Receivable, Accounts Payable, Inventory, letter writing and correspondence, mailing lists, data storage and report writing. Ready to run on any CP/M microcomputer system.* Ready to go to work for your business right now. We help business work better, faster, and smarter.

Our Product Line Includes:
- General Ledger
- Accounts Payable
- Inventory Control
- Letteright
- Accounts Receivable
- Payroll
- Analyst Data Storage/Report Writing
- NAD Mailing System
- QSORT Sort/Merge System

CP/M is a registered trademark of Digital Research.



*SSG business software runs on these and many other microcomputer systems. See your computer dealer for compatible hardware, or send for our brochure, Choosing A Computer For Your Business: SSG Software and Compatible Hardware.

Structured Systems

5204 Claremont Oakland, California 94618 (415) 547-1567

CIRCLE INQUIRY NO. 60

the Column

By Lyle Ronalds
SSI Far East Ltd.
Suite 201, Austin Centre, 21, Austin Avenue, Kowloon

This month's column will be taken from the February 1980 issue of "Readout," a newsletter published by Lyle Ronalds in Hong Kong.

The views and opinions stated in this column are those of the author and not necessarily those of the magazine and its staff.

BUSINESS MORALITY IN HONG KONG

Here's a story to touch your hearts; a giant U.S. electronic company undertakes to ship a sizeable consignment of components to a small Hong Kong end product manufacturer. The U.S. company accepts the order, and the letter of credit is opened.

Then, for reasons best known to himself, but shall we say because of a better price from a local U.S. customer, the U.S. component manufacturer decides not to deliver the goods, or to deliver only part to Hong Kong, or to deliver late. The U.S. component manufacturer thinks no more about it and goes to lunch.

In Hong Kong, the owner of the small factory does not go out to lunch, in fact, he has not had time to go out for lunch for some months. He has accepted an order from his own customer, and contracted to deliver at a set time at a set price. He has ordered all the other components necessary, and he has staffed up to meet production schedules. Then, he hears that his one major component consignment is not going to arrive, because his U.S. supplier has sold it to someone else. He has to renege on his promise to his customer, and loses a great deal of money and reputation in doing so. He has to resell at a loss all the other components he bought; and he probably also has to let a few people go who will have to find new jobs.
End of parable. Sound familiar?

Now, we'll stick our editorial neck out: The standard of business morality is higher in Hong Kong than it is in the U.S. While American public companies make a great show of being socially and morally responsible for their obligations to their employees, customers, communities, and suppliers, it seems to us, dealing with both Hong Kong and U.S. businessmen, that the Hong Kong executives are the ones who are practicing what the Americans are only preaching.

A Hong Kong businessman's word is his contract; lawyers get fat on digging for loopholes so that Americans can break theirs. A Chinese company will pay upon delivery of the goods, while in the States you're lucky if you get your money in sixty days.
The East, indeed, has a thing or two to teach the West about moral business conduct.□

UPDATE

TACTILE TELEPHONE POSSIBLE

Advances in robotics, coupled with recent technologies in developing artificial limbs, may lead to a new communication device, according to industry analysts. A phone equipped with a hand capable of transmitting human gestures, such as pointing, feeling and waving, is reportedly within the telephone state-of-the-art.

Dubbed Feel-a-Phone, the system could revolutionize conventional telecommunications by inserting expressive hand gestures into daily telephone conversations. Words alone lack the human emotion of a face-to-face meeting. Gestures transmit feelings and undertones that are not conveyed by voice, but could be transmitted through a Feel-a-Phone. An irate bill collector, for example, could convey his anger by violently shaking his forefinger at a late paying customer. A business deal conducted over the telephone could be topped off with a hearty handshake instead of "it's great doing business with you."

Recent developments in sensor and transducer technology, allowing for the production of durable, precise robot "hands," has reportedly caused many manufacturers of industrial robots to have backlogs stretching well into 1981. Manufacturing plants are turning to robots to perform a number of tasks including machining, welding, painting, and other dirty or undesirable jobs once performed by humans.

Some of the work corresponds to developments by the manufacturers of artificial limbs, who produce hands that look and act like the real thing. Research in the nuclear industry, where manipulator hands for weapons and power plants are widespread, relate to Feel-a-Phone where an operator holds material and operates machines by remote control, using robot manipulators guided by the operator's finger movement on control rings.

These techniques could be adapted in the design of a Feel-a-Phone, but would now be expensive. A Feel-a-Phone equipped with multiplexers to allow for transmission of about 60 different channels of position, would probably cost around $5,000. An additional voice channel would most likely be required, bringing the cost up to $10,000.

JOB FAIRS PROMOTE DP EMPLOYMENT

Newspaper headlines may trumpet the dangers of recession, but human resource directors of many national corporations don't believe a word of it. Personnel specialists for advanced technology companies almost unanimously repeat what today is a truism in employment recruitment: that engineers and technicians with the proper credentials are riding the crest of a "buyer's market."

So it is not surprising that an enterprising Minneapolis-based company, Business People Inc., has successfully put together a unique job mart, appropriately called Career Job Fair.

The promoters believe it is the first time a non-employment agency has put so many companies, engineers and other technical personnel together under one roof for match-making purposes. For companies, it is an opportunity to conveniently and inexpensively meet hundreds of qualified applicants; job shoppers in turn can select from a wide range of choices, and then quickly — and often anonymously — visit the desired companies.

The Career Job Fair was initiated in Minneapolis three years ago and more recently has been expanded into San Jose, California and Boston. Human resource directors are attracted to this type of recruiting program for a variety of reasons. Job Fair is a new and interesting approach to more traditional head hunting, according to Mike Dooley, personnel manager, Cardiac Pacemakers, Minneapolis. "It's the optimum way for applicants to look over a large number of companies in one day, and for companies to quickly meet a variety of prospects."

Charles Patterson, professional recruiter, Emerson Electric, St. Louis, uses the Job Fair both because the company has "many employment needs, and it's a public relations opportunity to dispell a misconception about Emerson. We are no longer in television and radio production, and use the Fair to acquaint the engineering market with our high technology and engineering work."

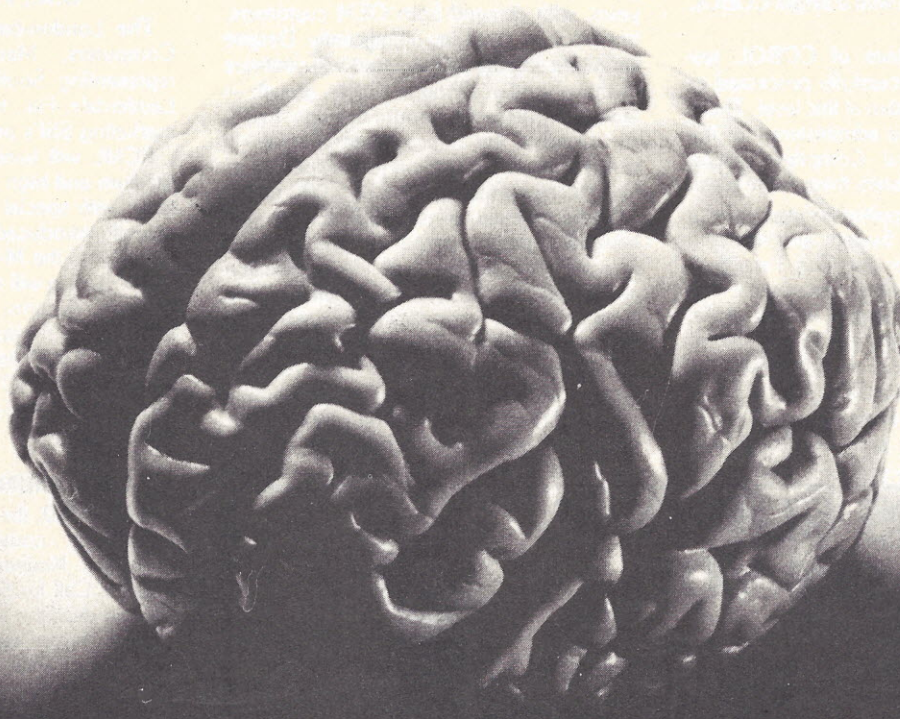
The concept may be unique but the novelty will wear out quickly without appropriate recruiting results. Last spring Control Data talked to approximately 200 at the company booth at the Minneapolis show and employed 13, including 11 exempt hires, according to Jim Lewis, Consultant for Corporate Staffing. "We didn't expect this number of hires and we're extremely pleased with the results." At the same show Honeywell made six major hires. At the Boston show Emerson recruited 12 and hired six and at Los Angeles employed seven. Signetics Corporation, Sunnyvale, did not reveal recruitment totals at the California show, but Dan Barryman, professional recruiter, considered his results to be "cost effective."

COBOL WINS GSA OK

Even though COBOL has been a standard industry language for some 20 years and is one of the languages of choice for government applications, it has only recently won official certification by the General Services Administration.

Developed by Micro Focus Ltd., London, England, and marketed in the United States by Micro Focus Inc., Santa Clara, California, CIS COBOL is a portable software system for compiling, debugging and executing COBOL programs.

THE ULTIMATE INFORMATION MANAGEMENT SYSTEM



The brain is the perfect information management system.

Like the brain, we at Micro-Ap specialize in the management of data.

Our software is the state of the art and is designed to efficiently store and instantly report the information you need for your business and personal needs.

From inventory control to mailing list management, Micro-Ap provides the most cost effective software available.

At the heart of our systems are Micro-Ap's unique indexing and reporting methods. You are not limited to single key retrieval. Information can be referenced by zip code, date, name, or any other indices required. Operation is "menu driven" and uses screen displays with all the instructions and error sensing that allow the novice to quickly learn the system and accomplish a multitude of tasks.

Micro-Ap's Newest Industry Innovation is the SELECTOR IV™ System, with the ability to make data file conversions, arithmetic computations, global search and replace, and full page report formatting.

This floppy and hard disc oriented system is upward compatible with SELECTOR III-C2™
It adds several dimensions to the world of information management.

Experience – SELECTOR™ has been around, and improving, longer than any other data base system in microcomputers.

See the innovative SELECTOR IV™ data base manager and the standard setting GLector, general ledger system at YOUR LOCAL COMPUTER STORE or contact:

Micro-Ap **9807 Davona Dr.** **San Ramon, Ca. 94583** **(415) 828-6697**	**Lifeboat Associates** **2248 Broadway** **NY, NY 10024** **(212) 580-0082** **telex 220501**

MICRO•AP

The Standard In Information Management Systems

CIRCLE INQUIRY NO. 45

Although it has only been available for a short time, it has become an industry standard for microcomputer COBOL operations with OEMs and end users. A major feature of the package is that it can be used with the RAM available in microcomputers (usually up to 64K bytes).

Since most COBOL applications require user participation as well as compactness, CIS COBOL is also interactive. This feature shortens program development time by allowing the user to debug programs in increments on a CRT screen. In real time applications, the interactive feature's run time option module displays a full screen of data on a CRT screen with a single COBOL language command.

Some 100,000 lines of COBOL test source code was successfully processed to achieve GSA certification at this level. These tests were devised and administered for the GSA by the Federal Compiler Testing Center, which administers them on request.

CIS COBOL is supplied to users with a compiler, Run Time System and interface module. The compiler and Run Time System are portable to new environments. The Run Time System is usually written in the assembler language of the target microcomputer which can be an Intel 8080, or 8085, a Zilog Z80 or a DEC LSI-11.

The CIS COBOL interface modules are specific to operating systems. These systems include CP/M for the Z80, the RT-11 for the LSI-11 and the ISIS-II for Intel's Intellec development systems. The interface module is the only CIS COBOL feature that is not completely interchangeable.

MINICOMPUTER REVENUES TO REACH $13 BILLION BY 1983

Throughout the 70s, the minicomputer market experienced dynamic growth rates in terms of both revenues and unit shipments. International Data Corp., Waltham, Mass., has closely watched this market over the last seven years. Its latest report concludes: Unit growth to decline steadily over the next few years, with demand from OEM customers, in particular, dropping significantly. Despite shrinking shipments, revenues will continue to show healthy, although slower, growth — increasing 29% per annum through 1983. At that point, the minicomputer market will be worth some $13 billion (with expectations for unit shipments reaching 235,000 units).

Some manufacturers (DEC for example) have taken steps to prepare for the expected downturn by cutting back on production and the building of new plant facilities. The strong revenue projections forecasted are due primarily to an expected increase in emphasis on service, software and add-on peripherals. In many cases, minicomputer manufacturers are enhancing their peripheral equipment operations to the point where many are actively marketing these products on an independent basis.

One major finding of IDC's research was significant increases in overseas revenues by a number of U.S. suppliers. Reasons include a generally higher average value per system for overseas shipments, and also international users taking advantage of the declining dollar in relation to their own currencies. By 1983, 39% of all revenues for U.S. manufacturers will be derived from overseas business with international shipments reaching 32% of the total.

SOUTHERN SYSTEMS NAMES U.K., FRANCE REP

The London-based firm of Peripherals, Computers, Memories & Leasing is representing Southern Systems Inc., Fort Lauderdale, Fla., in the U.K. and France in marketing SSI's printer systems.

PCML will handle the firm's full line of medium and high speed impact printer systems, with special emphasis on SSI's most recently introduced systems, the B-300, the B-600 and the M-200.

PCML also will market SSI's 2200 family of printer systems, based on 300, 600 and 900 line-per-minute drum printers; the 2550, a 1500 line-per-minute Charaband printer; and the CT1200 family, 600, 1000 and 1200 line-per-minute ChainTrain printer systems.

GATHERING SEEKS INPUT

Midcon/80, the Southwest's major high-technology convention and exhibition scheduled November 4-6 in Dallas, has issued a Call for Sessions. Manuscript submission deadline is July 25.

Each Midcon session will include three to five related papers covering, but not limited to, automotive electronics, communications, computers and microprocessors, consumer electronics, design automation, digital signal processing, electromagnetic compatibility, energy, instrumentation and measurement, manufacturing and testing, memories, office automation, petroleum electronics, reliability and quality control and semiconductor technology.

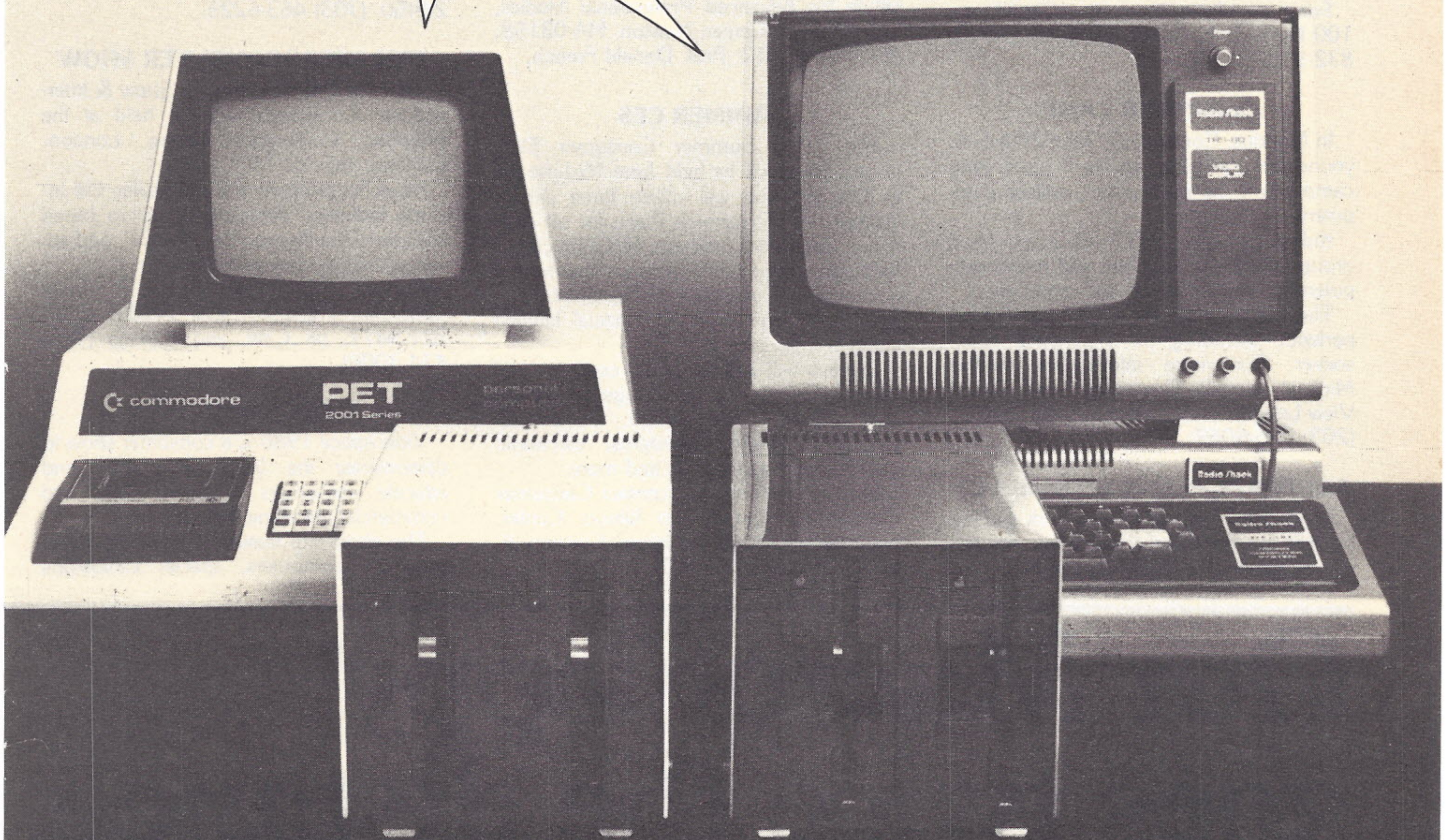
Contact Dale Literland, Educational Activities Manager, Midcon, 999 N. Sepulveda Blvd., El Segundo, CA 90245.

MORE IN EDUCATIONAL COURSEWARE

With an eye on the increasing use of personal computers in the home as an educational tool, Science Research Assoc. and Atari, both of Sunnyvale, CA, have combined their expertise in developing a multifaceted educational computer courseware package for use in the classroom and at home.

Science Research will develop software in such program areas as reading, languages, math, science and social studies, and Atari will market them in public and private schools from preschool through to the university level.

THANKS FOR THE MEMORIES.



Thanks to these inexpensive Pet and TRS-80 memories and disk drives you can instantly increase your personal computer's capacity, capability and versatility.

Expand your Pet.

Disk Drives: Choose either double density 400k or dual head double density 800k on-line disk drive. Both with DOS operating system including 17 additional BASIC commands and random or indexed sequential access.

Exceptionally fast 5,000 bytes per second (including verify) transfer speed lets you load 20k program in just 3 seconds. Model 8k must have Expandamem™ to operate disk drives. All other models are plug compatible.

400k only $1295. 800k only $1695.

Expandamem™: Lets you expand the size and scope of your programs. 16k, 24k, and 32k internal memory boards give you 8k of assembly language subroutines accessed via the USR command.

16k only $425. 24k only $525. 32k only $615.

Software: Complete library available. Write for details.

Expand your TRS-80.

Disk Drive: Single density dual head 400k on-line disk drive is completely compatible with TRS-80 DOS and uses own controller. Only $1295.

IMMEDIATE DELIVERY! ORDER TODAY!

COMPUTHINK
965 W. Maude, Sunnyvale, CA 94086.
(408) 245-4033.

Please send me the following:

☐ Pet Disk Drive ☐ 400k ☐ 800k
☐ Pet Expandamem™ ☐ 16k ☐ 24k ☐ 32k
☐ Pet Software (Info only)
☐ TRS-80 400k Disk Drive
☐ Name of my nearest Computhink dealer.

Total enclosed: $________

Check or Money Order only. Calif. Res. add 6% tax.
Freight pre-paid in continental U.S.
Dealer Inquiries Invited.

Name ________

Address ________

City ______ State ______ Zip ______

CIRCLE INQUIRY NO. 14

CALENDAR

SMALL COMPUTERS AND THE PROFESSIONAL

The Computers and Communications Committee of the Engineering Society of Detroit (ESD) will conduct a seminar on June 4 entitled "Small Computers and the Professional." The program is designed to meet the information needs of business professionals who are considering a purchase or expansion of a small computer system.

For more information contact Carol Lynn, 100 Farnsworth, Detroit, MI 48202, (313) 832-5400.

COMPUTER CAMP

In Moodus, Connecticut, June 29-July 4, youngsters can sign up for a week long camp where the main activity will be computers.

Kids from ages 10-17 will enjoy small group instruction and mini and microcomputers for ample "hands-on" experience.

The camp is for kids of all levels of experience including no experience whatsoever. For more information contact Michael Zabinski, Computer Camp, Grand View Lodge, Box 22, Moodus, CT 06469, (203) 795-9069.

SOFTWARE PLANNING SEMINAR

Polytechnic Institute of New York and the Institute for Advanced Professional Studies are presenting a three-day seminar for design, test, and diagnostic engineers and managers.

The seminar, entitled "Diagnostic Software: Planning and Design" will include design examples, lectures, informal sessions with instructors, as well as individual and group diagnostic programming sessions.

The seminar will be held July 14-16 at the Sheraton-Lexington Motor Inn, Lexington, Massachusetts. For details contact the Institute for Advanced Professional Studies, One Gateway Center, Newton, MA 02158, (617) 964-1412, Prof. Donald French.

SUMMER CES

The 1980 Summer Consumer Electronics Show will be held June 15-June 18 in Chicago and will utilize three exhibit facilities — McCormick Place for all consumer electronics products, McCormick Inn for audio components demonstration rooms and the Pick-Congress Hotel for demonstration rooms and suites for special interest audio components.

Exhibits will include audio compact and component systems, audio tape equipment and software, auto sound, video tape recorders, videodisc systems, electronic games, home computers and more.

For more information contact Consumer Electronics Shows, Two Illinois Center, Suite 1607, 233 N. Michigan, Chicago, IL 60601, (312) 861-1040.

MICROCOMPUTER INTERFACING

A two-week short course on the fundamentals of microcomputer interfacing will be offered by the Virginia Military Institute from July 14 through July 25.

This will be a hands-on laboratory oriented course which will feature the TRS-80 microcomputer (Level II with 16K).

For details contact Dr. Philip Peters, Dept. of Physics, VMI, Lexington, VA 24450, (703) 463-6225.

1980 MICROCOMPUTER SHOW

The 1980 Microcomputer Show & International Conference will be held at the Wembley Conference Centre, London, England, July 22-24.

Aspects examined and on display will include industrial applications, micro based commercial systems, micros in DP, and advanced micro system design.

Contact TMAC, 680 Beach St., Suite 428, San Francisco, CA 94109, (800) 227-3477, (in Calif. and Canada (415) 474-3000).

VIDEOSPACE 1980

Videospace 1980 is a consumer show to demonstrate the latest technology and educate consumers in home video and commercial electronic living.

The show will feature workshops, guest speakers, computers, special exhibitions

and displays of video hardware, security systems, personal computers and more.

Videospace 1980 will be held at Seattle Center, North Court, July 25-27. For details contact Michael Gaines, Rising Starr Productions, P.O. Box 17209, Seattle, WA 98107, (206) 682-7724.

COMPUTER GRAPHICS WEEK

Harvard Computer Graphics Week 1980 will be held July 28-August 1 at the Hyatt Regency Hotel in Cambridge, Massachusetts.

There will be discussions and examples of applications of business graphics and computer mapping in the commercial, educational, and governmental sectors, including displays of the most recent developments in graphic hardware.

Contact Kathy Devaney, Center for Management Research, 850 Boylston St., Chestnut Hill, MA 02167.

SILICON CONFERENCE

The Third International Conference on Neutron Transmutation Doping of Silicon will take place August 27-29 in Copenhagen.

Topics for the conference include transmutation physics, radiation defects, irradiation techniques, material properties and specifications, device applications, new devices and new materials.

Contact Motorola Inc., Semiconductor Group, P.O. Box 2953, Phoenix, AZ 85062.

FIFTH BIG YEAR FOR PCC

The Fifth Annual Personal Computing and Small Business Computer Show, PCC'80, will be held on August 21-24 at the Philadelphia Civic Center.

The show features exhibits and seminars highlighting all aspects of personal and small business computing.

For more information contact John Dilks, Personal Computing '80, Route 1, Box 242, Ward Rd., Mays Landing, NJ 08330, (609) 653-1188.

NEW JERSEY COMPUTER SHOW

The 1980 New Jersey Personal Computer Show and Fleamarket will be held September 27-28 at the Holiday Inn (North), at Newark International Airport (NJ Turnpike Exit 14).

The show will feature an indoor commercial exhibit area, a large outdoor fleamarket and user group meetings/forums on the TRS-80, PET, Apple, Heath and other popular systems.

For more information write NJPCS, Kengore Corp., 9 James Ave., Kendall Park, NJ 08824.

MINI/MICRO SHOW

The Mini/Micro Computer Conference and Exposition will be held at Brooks Hall/Civic Auditorium, San Francisco, California on Oct. 14-16.

For further information contact Robert D. Rankin, Managing Director, Mini/Micro Conference and Exposition, 32302 Camino Capistrano, Suite 202, San Juan Capistrano, CA 92675, (714) 661-3301.

PERSONAL AND BUSINESS COMPUTER SHOW

The Midwest Personal & Business Computer Show will be held at McCormack Place in Chicago from Thursday, October 16 through Sunday, October 19. Show hours are Thursday-Saturday: 11 a.m. to 9:30 p.m.; Sunday: 11 a.m. to 6 p.m.

For more information contact National Computer Shows, P.O. Box 678, Brookline Village, MA 02147, (617) 524-0000.

PERSONAL COMPUTER FAIR

The Northwest Computer Society and the Pacific Science Center will be holding the third annual Personal Computer Fair Nov. 8-9. The fair will be held in Seattle, WA. at the Pacific Science Center.

The theme of this year's Fair is "Hands On." The booths and exhibits will reflect this idea, with the public having access to as many computers and terminals as possible. There will be presentations for the beginner as well as the experienced professional.

For more information contact the Northwest Computer Society, P.O. Box 4193, Seattle, WA 98119.

BOOK REVIEWS

THE HOME COMPUTER REVOLUTION

By Ted Nelson. The Distributors, South Bend, IN. 224 pages, $2

Review by Susan Grace

Nelson's purposes in writing this book were to "explode" the myths surrounding computers and to explain the basics of computer knowledge. He succeeds in both areas.

In trying to explain why these myths exist, Nelson makes the observation that people are scared off by the mathematical sound of the term "computer"; this goes hand-in-hand with the more widely-held belief that not just anyone is capable of using a computer.

The fact that I have a limited knowledge of computers, yet could understand most of the ideas being explained attests that Nelson succeeds in his purpose of instructing the reader in the basics of computer knowledge. However, there are some areas that need more clarification for the beginner. The most difficult area, in terms of comprehension, is the section on computer languages, entitled "Tough, Optional Part." Another chapter that was slower-going concerned programming. A glossary of terms would have been helpful, not only here but in other parts of the book as well. It was difficult to go back and find the meaning of an unfamiliar term that popped up in the text, but this is partly due to the organizational structure of the book.

For the most part, *The Home Computer Revolution* is readable, because the book is written in a conversational style. Nelson's enthusiasm for his subject is obvious, and he exemplifies a point he makes early in his book: "The computer is a machine that brings out the kid in all of us." However, his style and enthusiasm can almost be described as juvenile at times. Example: "Surprise! There are thousands of different computer languages."

On an introductory level, this book is a good starting point for anyone interested in the who, what, when, where, why and how of computers.

After all, "a computer is simply a blank device whose purpose is chosen and whose steps are chosen by a human being," Nelson says. He also emphasizes that the true use of computers is for personal use.

Hey, did anyone just hear a myth explode?□

DESIGN OF TRANSISTOR CIRCUITS, WITH EXPERIMENTS

By Dr. Keats A. Pullen, Jr. Group Technology, Ltd., Check, VA. 508 pages, $12.95

Review by Michael Scott

Providing the background and explanations necessary to teach the reader the art of designing transistor circuits, this book is particularly useful in helping experimenters, amateurs, scientists, and engineers whose principal areas of activity are in fields other than electronics to develop an understanding of electronic circuits. Simple, valid explanations of the way solid state devices work and how they should be used are backed up with experiments that can be performed to verify the correctness of the statements.

The sound basic understanding developed will make the study of other books on electronics easier including those on microcomputers. Concepts rarely encountered in standard textbooks are shown to be important in the practical application of active devices.

Topics covered include basic theory, differences between linear resistances and the non-linear resistances on which solid-state devices are based, the relationships and applications of npn and pnp transistors, field effect devices, special purpose devices, evaluation of measurement devices, and special measurement problems.

Six appendices provide information about the Ebers-Moll model for an active device, useful circuits, instruments, and components; suggested supply sources for parts; additional experiments; and the characterization of active devices.□

OPERATING SYSTEMS — CONCEPTS AND PRINCIPLES

By John Zarrella. Microcomputer Applications. 140 pages, $6.95

Review by Alfred A. Adler, Ph.D.

In this first book of a series, Zarrella notes in the preface that books on computer software and hardware are "written on two levels — one for the computer science graduate student and one for the programmer attempting to learn a language on a specific computer." He feels that a need exists for a more intermediate level. He states, "This series is therefore dedicated to explaining some fundamental software engineering concepts, techniques and terms, and giving you, the reader, a feeling for the scope of the design problem."

In any technical field, the terminology is the first and quite possibly the biggest hurdle that must be overcome by the beginner. The computer field is probably the worst offender in this regard. Zarrella makes a valiant effort to cover as many buzz words as possible, putting them in boldface type as they arise in the text and devoting one-quarter of the book to a rather comprehensive glossary.

The book attempts to cover the entire broad range of operating systems, including multiprogramming, real time operations, multiuser, multiprocessing, system services, system support, scheduling, resource and memory management, input and output, file systems, and system security. The result, in only 100 pages, is a superficial description of what the words mean, and possibly who might want it and why. It certainly provides a wide angle view of the subject and gives the reader some feeling for the breadth of the problem, but at the same time leaves him with an empty feeling that he has really not learned much. The glossary covers 34 pages with some 230 entries.

The book should prove useful to anyone desiring a broad brush, quickly read treatment of the field of operating systems, without getting involved in anything heavy.□

Set fire to your paperwork.

Dispatch your paperwork with burning speed. With Spell Binder and a microcomputer you can set fire to just about any paperwork task in your office. Spell Binder is a software system that will drastically reduce the time you spend on paperwork, and ***increase*** your output.

WORD PROCESSING. Spell Binder is a word processing system you can learn to use in twenty minutes. But ease of use does not mean lack of powerful features. Compare Spell Binder's capabilities with any other word processing system. You'll be impressed. And Spell Binder doesn't stop with word processing.

MAILING LIST, MASS MAILING. Spell Binder features full mailing list and mailing label capability. And Spell Binder will merge your letter with your selected mailing list for personalized, tailored mass mailings. Sort and select by any key you want, including zip code.

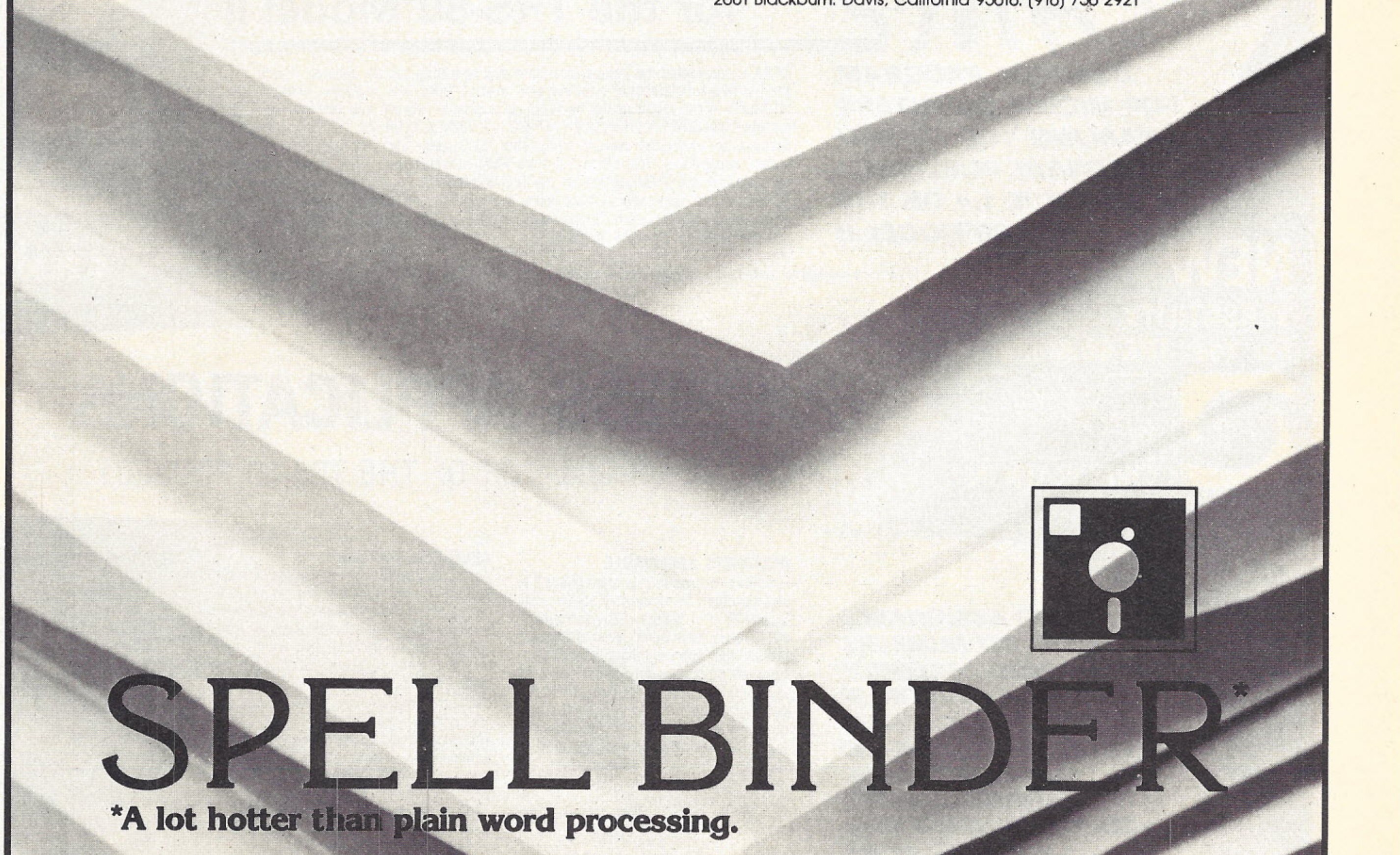
CUSTOM PROGRAMMING. Using Spell Binder's Macro Programming feature, you can shape your text to virtually any format, number the lines on legal text, create custom heads for running pagination, or do whatever you need for a fired up paperwork flow.

HOT, NEW, NOW. This new package is the best paperwork system you'll find on a microcomputer. It runs on computers with CP/M®, including the new Heath WH89. Spell Binder is available now. If it is not in your computer dealer's store for a demo, call us right away. Before your paperwork load turns into a real fire hazard.

CP/M is a registered trademark of Digital Research.
Spell Binder is a trademark of Testan Scientific Instruments.

CALIFORNIA PACIFIC COMPUTER CO.

2601 Blackburn, Davis, California 95616. (916) 756 2921



COMPUTER CLUB DIRECTORY

As a service to our readers, INTERFACE AGE is initiating a club directory to inform readers of clubs in their areas. To add your club to this directory, send club name, address and phone contact information to Club Editor, INTERFACE AGE, P.O. Box 1234, Cerritos, CA 90701.

AMATEUR COMPUTER GROUP OF NEW JERSEY
1776 Raritan Rd., Scotch Plains, NJ 07076
Jeff Kashinsky, President
(201) 536-1078

AMATEUR RADIO RESEARCH AND DEVELOPMENT CORPORATION
1524 Springvale Ave., McLean, VA 22101
Paul Rinaldo, President
(703) 356-8918

BOSTON COMPUTER SOCIETY
17 Chestnut Street, Boston, MA 02108
(617) 227-9178

NEW ENGLAND COMPUTER SOCIETY
P.O. Box 198, Bedford, MA 01730
Eric Johannson
(617) 562-6716

PHILADELPHIA AREA COMPUTER SOCIETY
P.O. Box 1954, Philadelphia, PA 19105
PACS Hotline
(215) 467-0177

ROCHESTER AREA MICROCOMPUTER SOCIETY
P.O. Box 90808, Rochester, NY 14607
Mike Ciaraldi
(716) 467-0177

ALAMO COMPUTER ENTHUSIASTS
4847 Castle Shield, San Antonio, TX 78218
David Samson
(512) 656-8469

COMPUTER HOBBYISTS GROUP OF NORTH TEXAS
P.O. Box 1344, Grand Prairie, TX 75051
Garrett Davis
(214) 559-2710

CRESCENT CITY COMPUTER CLUB
University of New Orleans
P.O. Box 1097
New Orleans, LA 70122
David Hughes
(504) 271-5540

DENVER AMATEUR COMPUTER SOCIETY
1380 S. Santa Fe, Denver, CO 80223
Mike Dmytrasz
(303) 697-5800

SOUTHEASTERN MICHIGAN COMPUTER ORGANIZATION
Box 02426, Detroit, MI 48202
S.E.M.C.O. Newsline
(313) 775-5320

UTAH COMPUTER ASSOCIATION
378 East 9800 South, Sandy, UT 84070
Lawrence Barney, President
(801) 571-9661

SPACE COAST MICRO COMPUTER CLUB
c/o Ray Lockwood
315 Inlet Avenue, Merritt Island, FL 32952
Ray Lockwood, President
(305) 452-2159

HOMEBREW COMPUTER CLUB
P.O. Box 626, Mountain View, CA 94042
Bob Reiling, President
(415) 967-6754

NORTHWEST COMPUTER SOCIETY
P.O. Box 4193, Seattle, WA 98104
(206) 284-6109

SAN DIEGO COMPUTER SOCIETY
P.O. Box 85137, San Diego, CA 92138
(714) 571-5550

SOLUS COMPUTER CLUB
1690 Woodside Road, #219
Redwood City, CA 94061
S. Sokolow
(415) 368-3331

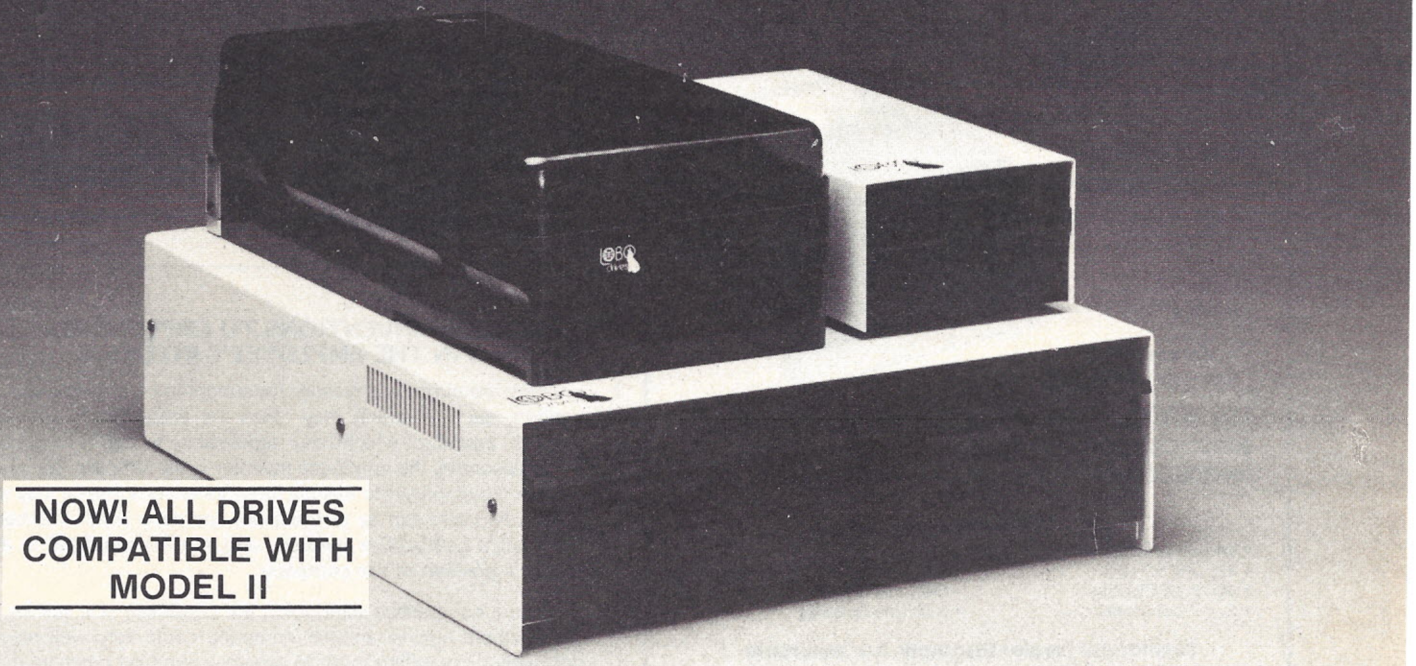
CONNECTICUT COMPUTER CLUB
c/o Leo Taylor
18 Ridge Court W., West Haven, CT 06516
Leo Taylor
(203) 933-5918

TCUG (TRS Computer User's Group)
P.O. Box 2235, Reston, VA 22090
Ron Hickey
(703) 241-2878

PROTEUS (International Processor Technology User's Group)
1690 Woodside Road, #219
Redwood City, CA 94061
S. Sokolow
(415) 368-3331

CIRCLE INQUIRY NO. 30

NEW FROM LOBO:

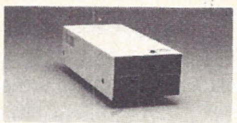


An Entire Family of Disk Drives for APPLE, TRS-80*, and S-100 Computers

Only LOBO DRIVES offers you an entire family of fully-compatible disk drives to select from. Whatever computer you're using, APPLE, TRS-80, or S-100, you can add a LOBO drive now, with the peace-of-mind of knowing there's a whole family of drives available when you're ready to expand.

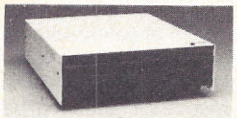
And every drive you order comes complete with chassis and high reliability power supply. Each drive is 100% calibrated, burned-in, and performance tested on either an APPLE, TRS-80, or S-100 computer before it's shipped. We are so proud of our drives . . . our quality, reliability, and performance, that we back-up every drive with a one year, 100% parts/labor warranty.

400 SERIES FLOPPY DISK DRIVES



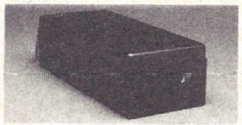
Meet our low-cost 5.25-inch mini drive that records data in either hard or soft sectored format. It is available in single or double density configurations, with a total storage capacity of 220K bytes.

800/801 SERIES FLOPPY DISK DRIVES



Here is our dual 8-inch Floppy disk memory unit. It records and retrieves data on standard 8-inch diskettes to provide 800K bytes of data storage unformatted, or 512K bytes in IBM format per drive. It is also available with double-sided, double-density capabilities, for a maximum storage capacity of 1.6 Megabytes.

7000 SERIES HARD DISK DRIVES



The latest member of our drive family, the Series 7000 is an 8-inch, 10 Megabyte Winchester Technology, hard disk drive. It is fully hardware/software compatible and comes complete with disk controller. Now you can have the convenience, speed, reliability, and all the storage capacity you need.

Call or write for the complete LOBO DRIVES story. Find out just how competitively priced a quality drive can be.

Quantity discounts available –
Dealer inquiries invited.

Yes, I want to know more about LOBO Drives and what they can do. Send me information on:
☐ TRS-80 ☐ APPLE ☐ S-100

☐ 5 1/4-in. Floppy drive
☐ 8-in. Winchester hard disk, 10 Mbyte drive
☐ 8-in. Floppy drive Single sided Double sided
☐ Double density expansion interface

Name ____
Company ____
Address ____
City ____ State ____ Zip ____
Phone No. ____
If dealer, provide resale no. ____

*TRS-80 is a registered trademark of Radio Shack, a Tandy Company.

LOBO drives INTERNATIONAL

935 Camino Del Sur
Goleta, California 93017
(805) 685-4546

"CAN YOU REALLY AFFORD TO PAY LESS?"

Jurisprudent Computerist

By Leonard Tachner
Attorney-at-Law

DISCLOSING INVENTIONS TO EMPLOYERS: WHY IT'S IN THE EMPLOYEE'S INTEREST

There are numerous reasons for submitting a disclosure on an invention, each of which may affect the employee in one way or another, and all of which have significance to the employer.

Disclosure by the employee-inventor is the simplest and cheapest way for the employer to identify those inventions that he may want to patent. A patent can be a significant company asset. It enhances a company's competitive position, and can be beneficial to an employee's position in the company.

From a competitive standpoint, the possibility of a suit for patent infringement against another company could very well result in a competitor expanding funds to design around the patented development rather than taking the risk of infringing the patent. Thus, additional expense is incurred by such a competitor in designing around, preventing him from getting a free ride on a company's investment and technical expertise. This advantage may help to secure an order or boost sales of a product related to an employee's area of technical expertise.

Another advantage is the defensive trading position against other patent owners. Cross-license agreements may allow a company to enter the market at a lower cost than might otherwise be possible. The trading value of patents, particularly in new technologies, can result in lower royalties or in the elimination of patent royalties that might otherwise have to be paid. Similarly, the company may be able to enter a market that it might not otherwise have entered without a very substantial investment for designing around the patents of others. This could easily spur development in a technology to which the employee's expertise relates.

Patents provide protection to the company against the issuance of a patent to another on the same development. This is usually referred to as protecting the right to use. Delaying disclosure could result in someone else getting a patent. That would not be in either the employee's or the company's interest.

Patents also provide a basis for royalty income. There are occasions when it is economically advantageous to license someone else to make, use, or sell the product on which a company has a patent. When this is done, it is usually in a product area in which the company has not traditionally participated and has not made the investment required to enter that field. Thus, patent licensing gives a company the opportunity to earn income by way of royalties in a product area in which it might not otherwise have participated.

Finally, the establishment of a portfolio of patents on a particular product, or in a particular technology, makes it more difficult for a competitor to design around. The degree of commercial success in a particular product area may be increased if a number of patents are obtained on various inventions used in a product. This increases the economic burden to which competitors are put, in order to compete with a company without running the risk of infringing a patent.

Thus, in the best interests of all concerned, employees should submit disclosures on their inventions. The name of the game is competition. A company can legally obtain a competitive edge through the patent system, and can also offer increased opportunities for its employees.□

new from...
HARDSIDE
6 SOUTH ST., MILFORD, NH 03055
ATARI®
A Warner Communications Company
ATARI 800
ATARI 400
ATARI 825
ATARI 410™
ATARI 850 INTERFACE MODULE
ATARI CX852 MEMORY MODULE 8K RAM
ATARI CX853 MEMORY MODULE 16K RAM
EDUCATIONAL SYSTEM
BASIC
BASKETBALL
ATARI 810
ATARI 820
ATARI 830
HARDSIDE™
TO ORDER TOLL-FREE
1-800-258-1790
(In N.H. call 673-5144)
ATARI 800 (list 1080.00) 875.00
ATARI 400 (list 630.00) 515.00
810 DISK DRIVE (699.95) 565.00
820 40-COLUMN PRINTER (list 599.95) 470.00
410 PROGRAM RECORDER (list 89.95) 65.00
JOYSTICK CONTROLLERS . . 18.95
PADDLE CONTROLLERS . . . 18.95
ATARI 850 INTERFACE . to be announced
ATARI 830 MODEM . . . . to be announced
ATARI 825 PRINTER . . . . to be announced

AL BAKER'S GAME CORNER

By Al Baker

Raise your hand if you have Programmer's Aid #1 in your Apple II. Have you tried to use its music? Well, today you will. This month, our program is called PLAYER PIANO. It uses the musical notation developed by Dick Ainsworth of The Image Producers for Bally BASIC and APF BASIC.

Player Piano isn't a game, toy, or even pure entertainment. It is a serious attempt at using a computer to play music. But it is fun, and it does show off some of the musical potential in Apple's Programmer's Aid #1.

MUSICAL NOTATION

Player Piano uses the basic structure of the musical notation. Programmer's Aid #1 doesn't support multiple voices, a large number of octaves, or the creation of note envelopes. The notes of the C major scale are 1 through 7. On the Apple II, the octave immediately below middle C is the default.

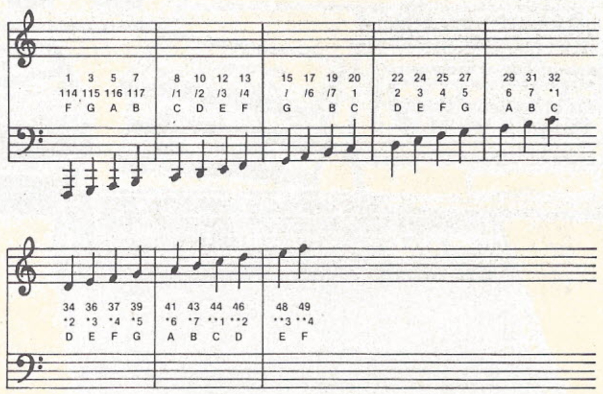



Figure 1. The musical scale: First row of numbers is Programmer's Aid #1's own internal pitch notation. The second row is our song notation.

Figure 1 shows the four octaves supported by Programmer's Aid #1. The bottom row written between the staves is the standard notation. By itself, you can't play the tune. You need to see each note on the staff to know its octave. The upper row is the pitch notation needed by Programmer's Aid #1. Translating a song from sheet music to this notation is time consuming.

Here are three songs written using Player Piano's notation. Zeros extend a note and spaces are rests. See if you can recognize them before reading on:

```
3212333 222 355
144557641445560
660367030*17*1*2*107
```

The first is "Mary Had a Little Lamb." Next is "Lemon Tree." Last is "O Come, All Ye Faithful."

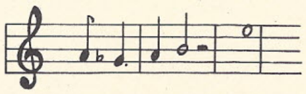
Here is a summary of the rules for the basic notation:

- 1 through 7 are the notes of the C major scale
- use * to go up an octave

- use / to go down an octave
- use - for a flat note
- use + for a sharp
- use 0 to extend a note
- use a blank for a rest

USING PLAYER PIANO

I have provided several songs for you to play. Run Player Piano and it will ask you to type in a song. Use as many lines as necessary to enter it. When done, enter an empty line. After a brief pause, your song will play. You can play it as many times as you want, or you can enter a new song.

It is easy to create your own songs. That's how I got some of those here. If you can't read sheet music, you may need someone's help. Find the shortest note in the piece. Suppose it is an eighth note (which looks like this ♪ or ♫). Then quarter notes will have one zero after them and half notes will have three. Whole notes will have seven. Here is the translation of this piece:

6-500607000 *30000000

If the piece sounds too fast or slow, change program line 230. The lower the number, the faster the song. And don't forget sharps and flats. Your song won't sound right if you forget to consider the key it is written in.

THE PROGRAM

Lines 150 through 180 set up the constants used by Programmer's Aid #1. MUSIC is the address of the music subroutine, PITCH is where we place the note to be played, and TIME is where we set the note's length. Line 180 sets up the proper timbre of the music. Read the Programmer's Aid document for a description of these locations.

IN$ is the person's input string. A$ is where the program keeps the entire song. Later, we convert the player piano's notation into pitch and time. These will be kept in the P and T arrays.

TT is the duration of our shortest note. Change this to speed up or slow down the song. The B array contains the pitch values for the C major scale.

From lines 340 through 400, the program builds the user's song. Lines 440 through 480 play the song as many times as desired. The remainder of the program converts the player piano musical notation into the computer's.

For each note, we begin by assuming it will be the smallest length, not a flat or sharp, and on the default octave. If a "/" is found, we subtract the 12 pitch units of the tempered scale. If a ":" is found, we add the 12 half steps of the tempered scale. This handles changing octaves. Likewise, "+" or ";" creates a sharp by raising the note a half step and "-" creates a flat by lowering the note a half step.

Each time we find a zero in the string, we lengthen the note by one time unit. Rests are handled by using a pitch of zero. The loop on lines 960 through 1080 then plays the converted song.

ON YOUR OWN

This program doesn't have a music editor in it. It doesn't even save your song. But it does have the building blocks. If you enjoy playing music on your computer, it's now up to you.□

LISTING 1

```
100 REM ...PLAYER PIANO...
110 REM
120 REM
130 REM  DEFINE VARIABLES
140 REM
150 MUSIC=-10473
160 PITCH=767
170 TIME=766
180 POKE 765,32
```

```
190 DIM A$(200),P(200),T(200),B(7),H$(1),IN$(40)
200 REM
210 REM  SET TEMPO, NOTES
220 REM
230 TT=25
240 B(1)=20
250 B(2)=22
260 B(3)=24
270 B(4)=25
280 B(5)=27
290 B(6)=29
300 B(7)=31
310 REM
320 REM   ACCEPT USER'S SONG
330 REM
340 A$=""
350 PRINT "ENTER SONG. I WILL ACCEPT INPUT UNTIL"
360 PRINT "YOU ENTER AN EMPTY LINE."
370 INPUT "ENTER SONG LINE:",IN$
380 IF IN$="" THEN 440
390 A$( LEN(A$)+1)=IN$
400 GOTO 370
410 REM
420 REM  PLAY THE SONG AS OFTEN AS DESIRED
430 REM
440 GOSUB 560
450 INPUT "PLAY IT AGAIN (Y,N)?",IN$
460 IF IN$="Y" THEN 440
470 IF IN$="N" THEN 340
480 GOTO 450
490 REM
500 REM
510 REM  PLAY THE MUSIC STRING
520 REM
530 REM
540 REM  FIRST, CONVERT TO THE CORRECT NUMBERS FOR PITCH AND TIME"
550 REM
560 OCT=0
570 MOD=0
580 J=1
590 FOR I=1 TO LEN(A$)
600 T(I)=TT
610 H$=A$(I,I)
620 IF H$="/" THEN OCT=OCT-12
630 IF H$="*" THEN OCT=OCT+12
640 IF H$=":" THEN OCT=OCT+12
650 IF H$="+" THEN MOD=MOD+1
660 IF H$=";" THEN MOD=MOD+1
670 IF H$="-" THEN MOD=MOD-1
680 IF H$="0" THEN 770
```

```
690 IF ( ASC(H$)> ASC("0"))*( ASC(H$)< ASC("8")) THEN 830
700 IF H$#" " THEN 910
710 REM
720 REM MUSICAL RESTS HANDLED HERE
730 P(J)=0
740 GOTO 880
750 REM
760 REM LONGER NOTES HANDLED HERE
770 T(J-1)=T(J-1)+TT
780 IF T(J-1)>255 THEN T(J-1)=255
790 GOTO 910
800 REM
810 REM NOTES HANDLED HERE.
820 REM NOTE = BASE NOTE + FLAT/SHARP. MODIFIER + OCTAVE. CHANGE
830 K= ASC(H$)-128-48
840 K=B(K)+MOD+OCT
850 IF K>50 THFN K=50
860 IF K<1 THEN K=1
870 P(J)=K
880 J=J+1
890 MOD=0
900 OCT=0
910 NEXT I
920 REM
930 REM PLAY THE CONVERTED DATA
940 REM
950 J=J-1
960 FOR I=1 TO J
970 POKE PITCH,P(I)
980 POKE TIME,T(I)
990 CALL MUSIC
1000 NEXT I
1010 RETURN
```

LISTING 2: Marine's Hymn

```
1350505050        604060405006
500*15034         50*1760406*100
505042001000      500000135050
001350505050      5050500*15034
500*150345050     500050006000
4200100000*17     7000*1000
```

LISTING 3: Amazing Grace

```
20500075700060      70*2007*275000
5000302000205000    2300553200020
75700060*2000       5000757000605000
```

LISTING 4: Bagpipes

```
405654-70*2*106     46*2*164505505
406654502300        46*2*16460*1
405654-70*2*106     *20*3*4*2*1
*406654504401       654605400
46*2*164606605
```

LISTING 5: Frankie and Johnny

```
1236053010100000
1236053010000010
456*10*260*10
*100067*10*1*17060
503030-3020000000
```

blanks → 6565001000000000000

LISTING 6: Greensleeves

```
30500060700+170
6000+402003+40500030
300+230+4000+20/700030
500060700*1706000+40
2003+40500+430+200+120
300030300000*200000*200
+*1706000+402003+40
500030300+230+4000
+20/700000*200000*200
+*1706000+402003+400
500030+200+1203000003000
```

Diagnostics I for CP/M* & TRSDOS#

Someday your computer is going to break; even the most reliable computer systems "go down". Often, finding exactly what is wrong can account for the most time consuming part of repairing the system, and the longer the system is down, the more money you lose.

DIAGNOSTICS I is a complete program package designed to check every major area of your computer, detect errors, and find the cause of most common computer malfunctions, often before they become serious. For years, large installations have run daily or weekly diagnostic routines as a part of normal system maintenance and check-out procedures.

DIAGNOSTICS I is designed to provide that kind of performance testing for 8080/Z80 micro computers.

DIAGNOSTICS I will really put your system through its paces. Each test is exhaustive and thorough. The tests include:

- Memory Test
- Disk Test
- Printer Test
- CPU Test (8080/8085/Z80)
- CRT Test

To our knowledge, this is the first CPU test available for 8080/Z80 CPU's. Many times transient problems, usually blamed on bad memory, are really CPU errors.

A good set of diagnostics is an indispensable addition to your program library even if your system is working fine. Hours have been wasted trying to track down a "program bug" when actually hardware was to blame!

DIAGNOSTICS I also allows you to be confident of your system. This can be critical when file merges or sorts and backups are involved. You want to be as sure of your computer as possible during these critical times. Running DIAGNOSTICS I prior to these and other important functions helps to insure that your system is operating at peak performance.

DIAGNOSTICS I is supplied on discette with a complete users manual.

DIAGNOSTICS I: $50.00
Manual: $15.00

Requires: 24K CP/M; 16K disc for TRS-80
formats: CP/M 8" SOFT SECTORED, NORTHSTAR CP/M AND TRS-80 DOS

All Orders and General Information:
SUPERSOFT ASSOCIATES
P.O. BOX 1628
CHAMPAIGN, IL 61820
(217) 344-7596
Technical Hot Line: (217) 384-0847
(answered only when technician is available)

Give your computer a "physical" today!

VISA · master charge

SuperSoft
First in Software Technology

CIRCLE INQUIRY NO. 62

*CP/M REGISTERED TRADEMARK DIGITAL RESEARCH
#TRSDOS, TRS-80 TRADEMARKS TANDY CORP

THE MICRO-MATHEMATICIAN

By Dr. Alfred Adler

NUMERICAL INTEGRATION OF TRAJECTORY EQUATIONS

Review of Integration

Last month we explored the general subject of integration. We used a very simple equation to demonstrate the analytical process of integration, and pointed out that if an equation is not available, as in the case of experimental data, any necessary integrations must be performed numerically.

The subject of numerical integration was examined and two of the simplest forms, the Trapezoidal Rule and Simpson's Rule, were looked at in detail. Program NUMERINT, which facilitates comparisons between these two methods, was presented along with a number of sample runs.

It was concluded that, in the majority of cases, Simpson's Rule gives more accurate results in much less time than the Trapezoidal Rule. Under certain circumstances, however, Simpson's Rule may present no advantage and might possibly even introduce difficulties.

As shown in last month's column, simple equations can usually be integrated analytically. The result, of course, is another equation. Everything is very convenient and tidy. What was not stated last month, however, is that there exist many types of neat and tidy equations that cannot be integrated analytically. Only certain forms are integrable, and an equation not fitting the limited number of possible molds is generally not integrable. Having shown in last month's column how to use numerical integration to find the area under a curve, it would be instructive to continue by showing how to solve a differential equation numerically.

Trajectory Equations

Among the many types of differential equations that are not easily integrable are the equations describing the trajectory of an object under the influence of gravity, thrust, and aerodynamic forces. If aerodynamic forces are either ignored or simulated in a simplified manner, and if one or two of a variety of other simplifications are made, the equations of motion can be integrated analytically.

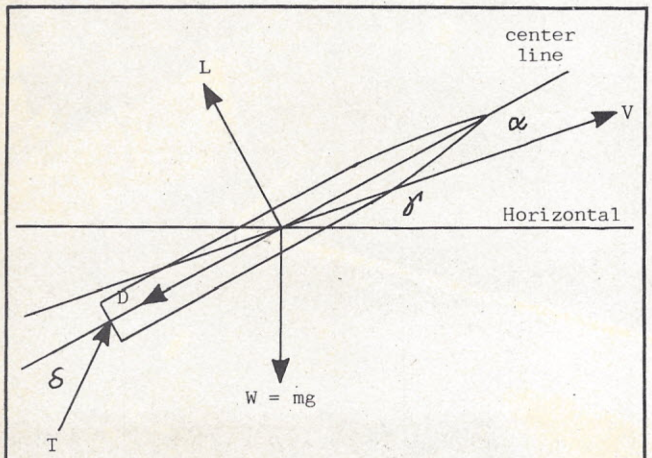


Figure 1. Atmospheric forces on an object in flight.

Whether this results in a neat little package or a mess depends on the details of the simplifications. We are concerned with analytical integration of these equations since we want to be able to determine the accuracy of the numerical integration procedure used. We must therefore abridge the equations to the point where they can be integrated analytically. In this article, we shall integrate two abridged versions of the trajectory equations. One is very abridged, the other about as little as can still be reasonably handled analytically. A determination of the accuracy of the numerical equation will be made and the difficulties and possible remedies explored.

Figure 1 shows the forces on an object in flight in the atmosphere, under the influence of gravity, thrust, and aerodynamic forces. The equations of motion, parallel and perpendicular to the flight path, along with the required auxiliary equations, are shown in figure 2. These equations already include many simplifying assumptions, among them a spherical nonrotating earth, two-dimensional motion, stepwise constant thrust, aerodynamic forces that act at the vehicle center of mass, etc. These equations are presented primarily to give the reader an idea of what is involved in determining the trajectory of even a non-winged vehicle (a winged vehicle is far more complex), even under the simplifying assumptions stated. These equations, of course, must be integrated numerically.

$$m\frac{dV}{dt} = T\cos(\alpha+\delta) - mg\sin\gamma - D$$

$$mV\frac{d\gamma}{dt} = L + T\sin(\alpha+\delta) - mg\cos\gamma + \frac{mV^2\cos\gamma}{R+h}$$

$$T = I_{sp}g\frac{dm}{dt}$$

$$g = \frac{GM}{(R+h)^2}$$

$$m = m_o - \frac{dm}{dt}t$$

$$r = R + \int V\sin\gamma\, dt$$

$$\text{Range} = R\int \frac{V}{r}\cos\gamma\, dt$$

where G = Universal gravitational constant
M = mass of Earth
R = radius of Earth
r = distance from center of Earth
h = altitude above surface of Earth
I_{sp} = specific impulse of propellant
T = thrust
V = velocity
m = mass of vehicle
γ = flight path angle with local horizontal
D = Drag
L = Lift
subscript o = initial condition

Figure 2. Equations of motion, parallel and perpendicular to the flight path.

Let us first abridge the equations until we have a set that are easily integrated analytically. We make the following assumptions.

1. Lift and drag equal zero.
2. Thrust equals zero.
3. The flight range is small compared to the radius of the earth. This permits the assumption that the earth is flat, which implies that g is constant. This limits the flight altitudes to under about 100 miles.
4. The flight path is vertical.

Having made these assumptions, the equations reduce to those presented in figure 3. These are, of course, the familiar, so-called falling body equations given in every high school physics text. They are, however, valid for motion in either direction.

$$\frac{d^2h}{dt^2} = -g \qquad (1)$$

$$V = \frac{dh}{dt} = v_o - gt \qquad (2)$$

$$h = h_o + V_o t - \frac{1}{2} gt^2 \qquad (3)$$

Figure 3. Simplified flight path equations.

Since the equations for velocity and altitude in figure 3 were obtained analytically by successive integration of the first equation, the results they give are accurate and will be considered 'correct' within the limits of our simplifying assumptions. In order to perform numerical integration on the first equation, it must be recast. The d in all these equations implied an infinitely small increment. Thus dh/dt means an infinitely small increment in h, divided by an infinitely small increment in t. This quotient represents the rate of change of altitude with respect to time and is, of course, the vertical velocity. We cannot represent an infinitely small quantity on a digital computer. Therefore the equations must be written in a form that eliminates the need for infinitesimals. We can, of course, always take a finite number of finite steps instead of an infinite number of infinitesimal steps. The larger the finite size steps, however, the greater the error in the results. That is exactly what we will do. Using the upper case delta to represent a finite increment, we can rewrite the equations in figure 3.

Equation 1 is rewritten as equation (4) in figure 4. The latter states that the finite increment in vertical velocity equals minus g times the finite increment in time. This can be handled by a digital computer. Using the equations of figure 4, we proceed stepwise. Equation (4) yields the increment in velocity. By adding that to the previous velocity, we obtain an updated value, equation (5). Using this updated velocity we obtain the increment in altitude from equation (6). This, of course, assumes that V is constant during each time increment, and that is where the error arises. If the increment is infinitesimal, as it is in the analytic solution, this is correct, and we have zero error. But if the increment is larger (finite), it is in general not correct, and we have an error. Obviously, the smaller the steps taken (timewise, in this case) the smaller the error. Of course, in the event that the velocity actually is constant during the time increment, the numerical integration would be exact. It turns out that the increment in velocity given by equation (4) is exact, regardless of step size, since g is constant. It is for this reason that equation (5) yields exact results, regardless of step size (see the sample runs). Finally, adding the increment in altitude from equation (6) to the previous altitude, we obtain an updated value (see equation (7)).

$$\Delta V = \Delta \frac{dh}{dt} = -g \Delta t \qquad (4)$$

$$V_{n+1} = V_n + \Delta V \qquad (5)$$

$$\Delta h = V \Delta t \qquad (6)$$

$$h_{n+1} = h_n + \Delta h \qquad (7)$$

Figure 4. Finite increment of simplified flight path equations.

Program VERTRAJ1

Program VERTRAJ1 presents tabulated values of V and h from equations (5) and (7) for comparison against the values from equations (2) and (3). The computational sequence discussed above is followed exactly. The user is asked to input initial values of altitude

and velocity, and then must choose the integrating interval, that is the time increment, and the print interval. A listing of program VERTRAJ1 is given in figure 5.

```
10 REM|||||||||||||||||||||||||||||||||||||||||||||||||||||||||||
20 REM
30 REM|||||||||||||| P r o g r a m   V E R T R A J 1 ||||||
40 REM
50 REM||||||||||||| Written by - Alfred A. Adler, Ph.D. |||
60 REM
70 REM||||||||||||||| Version 1.0 - January 1980 |||||||
80 REM
85 DIM F(366)
88 P2=6.2831853
95 !
98        REM ****** INPUT DATA ******
100 INPUT"Initial altitude ?",H0
110 INPUT"Initial velocity ?",V0
120 INPUT"What is the integrating interval ?",T1
130 INPUT"What is the print interval ?",T2
135 !
140 G=32.174
145       REM ****** INITIALIZE ******
150 H=H0\V=V0\Z=0
160 !TAB(6),"T",TAB(14),"H (N.I.)",TAB(26),"H (ANAL)",TAB(38),
170 !"V (N.I.)",TAB(50),"V (ANAL)"
175       REM ****** TRAJECTORY COMPUTATION ******
180 FOR T=0 TO 1000 STEP T1
190       REM ****** ANALYTIC ******
200 V9=V0-G*T \REM V9=V (ANAL)
210 H9=H0+V0*T-G/2*T^2 \REM H9=H (ANAL)
220 IF T/T2<>INT(T/T2) THEN 250
230 !%9F2,T,%12F2,H,H9,V,V9
240 IF Z=1 THEN EXIT 95
245       REM ****** NUMERIC ******
250 V3=-G*T1 \REM V3=DELTA V (N.I.)
260 V=V+V3
270 H3=V*T1 \REM H3=DELTA H (N.I.)
280 H=H+H3
290 IF V<0 THEN IF H<0 THEN Z=1
300 NEXT
READY
```

Figure 5. Program VERTRAJ1

Sample Runs on VERTRAJ1

Sample runs on program VERTRAJ1 are presented in figures 6 and 7. In figure 6, we integrate every 1 second. The numerically integrated values of V are exact for reasons already discussed. The numerical values of h, however, contain considerable error. In an effort to reduce this error, we integrate on a much smaller time increment, as shown in figure 7. These values of h are much more accurate, possibly satisfactory for some purposes. But at what cost? The run in figure 6 took 9 seconds, but the run in figure 7 took 100 times that long (16 minutes). That is intolerable if many runs are to be made.

```
Initial altitude ?500
Initial velocity ?1000
What is the integrating interval ?1
What is the print interval ?10

     T      H (N.I.)    H (ANAL)    V (N.I.)    V (ANAL)
    .00      500.00      500.00     1000.00     1000.00
  10.00     8730.43     8891.30      678.26      678.26
  20.00    13743.46    14065.20      356.52      356.52
  30.00    15539.09    16021.70       34.78       34.78
  40.00    14117.32    14760.80     -286.96     -286.96
  50.00     9478.15    10282.50     -608.70     -608.70
  60.00     1621.58     2586.80     -930.44     -930.44
  70.00    -9452.39    -8326.30    -1252.18    -1252.18

Initial altitude ?
STOP IN LINE 100
READY
```

Figure 6. Program VERTRAJ1 integrating every 1 second.

```
Initial altitude ?500
Initial velocity ?1000
What is the integrating interval ?.01
What is the print interval ?10

     T      H (N.I.)    H (ANAL)    V (N.I.)    V (ANAL)
    .00      500.00      500.00     1000.00     1000.00
  10.00     8889.69     8891.30      678.26      678.26
  20.00    14062.00    14065.20      356.52      356.52
  30.00    16016.89    16021.70       34.78       34.78
  40.00    14754.37    14760.80     -286.96     -286.96
  50.00    10274.45    10282.50     -608.70     -608.70
  60.00     2577.15     2586.80     -930.44     -930.44
  70.00    -8337.44    -8326.30    -1252.15    -1252.18

Initial altitude ?
STOP IN LINE 100
READY
```

Figure 7. Program VERTRAJ1 with smaller time increment.

Recall that the main source of error is that the increment in h is computed in equation (6) on the assumption that V is constant during the integration interval. This V incidentally is the updated value, which means that it is the value at the end of the time interval. That explains why h is too low on the way up and also on the way down. Suppose we save the old value of V, that is the value at the end of the previous interval, average that with the value from equation (5), and use the average value in equation (6). This should improve the accuracy of the numerical integration.

Program VERTRAJ2

Program VERTRAJ2 is simply VERTRAJ1 modified as discussed above. In the interest of economy, we present only that part of the listing below REM******NUMERIC******, see figure 8. The remainder of the listing is identical to that shown in figure 7.

```
245       REM ****** NUMERIC ******
250 V3=-G*T1 \REM V3=DELTA V (N.I.)
255 V4=V
260 V=V+V3
265 V5=(V4+V)/2
270 H3=V5*T1 \REM H3=DELTA H (N.I.)
280 H=H+H3
290 IF V<0 THEN IF H<0 THEN Z=1
300 NEXT
READY
BYE
+
```

Figure 8. Program VERTRAJ2

Sample Runs on VERTRAJ2

We repeat the run of figure 6 using the modified program, see figure 9. It turns out that this is the only sample run required which, incidentally, took only 11 seconds. The modification permits program VERTRAJ2 to give numerical integration results that are exact, regardless of how large an integrating interval is used, as subsequent runs demonstrated. How can this be? Examination of the equations reveals that the velocity is a linear function of time. Therefore, multiplying the average velocity over the time interval, by the time interval, yields the exact value for the change in altitude. How fortunate.

```
Initial altitude ?500
Initial velocity ?1000
What is the integrating interval ?1
What is the print interval ?10

     T      H (N.I.)    H (ANAL)    V (N.I.)    V (ANAL)
    .00      500.00      500.00     1000.00     1000.00
  10.00     8891.30     8891.30      678.26      678.26
  20.00    14065.20    14065.20      356.52      356.52
  30.00    16021.70    16021.70       34.78       34.78
  40.00    14760.80    14760.80     -286.96     -286.96
  50.00    10282.50    10282.50     -608.70     -608.70
  60.00     2586.80     2586.80     -930.44     -930.44
  70.00    -8326.30    -8326.30    -1252.18    -1252.18

Initial altitude ?
```

Figure 9. Program VERTRAJ2 using run of figure 6.

Less Abridged Equations

Our good fortune with the previous set of equations was due to the very severe constraints put on the trajectory. Exact solutions by numerical integration are not an everyday occurrence. This time we will formulate equations for a more realistic trajectory, while still remaining sufficiently constrained that they can still be solved analytically. This is an absolute requirement if we are to determine the accuracy of the integration.

Again starting with the equations in figure 2, we modify the previous assumptions as follows.

1. Lift equals zero, but drag = K*V*Xdot, where K is the product of the drag coefficient, the reference area, and one-half the air density. Xdot is the horizontal component of the velocity.
4. The flight path is non-vertical, see text.

With these assumptions, the equations reduce to those presented in figure 10. They are significantly more complex than the previous set. The reason incidentally for the drag assumption, number 1

above, is that the correct formulation D = K*V^2 produces a very messy analytic solution. For trajectories with flight path angles not exceeding 45°, Xdot is not too radically different from V, and the results are surprisingly good. The assumption of constant air density, of course, requires that either the trajectory be entirely at high altitude, or that the altitude change be small. This again requires a flat trajectory. At lower altitudes the accuracy can be improved considerably by using a value for air density equal to that at the initial altitude minus one third of the difference between the value at the initial altitude and the value at the apogee (the highest point) of a trajectory computed letting density equal the value at the initial altitude. This is noted in the heading of the listing of program TRAJ/GD1.

$$\frac{d^2x}{dt^2} = \frac{-KV\dot{x}}{m}\frac{\dot{x}}{V} \qquad (8)$$

$$\frac{d^2h}{dt^2} = \frac{-KV\dot{x}}{m}\frac{\dot{h}}{V} \qquad (9)$$

$$\dot{x} = \frac{dx}{dt} = \frac{\dot{x}_o}{1 + \frac{K}{m}\dot{x}_o t} = \text{Xdot} \qquad (10)$$

$$\dot{h} = \frac{dh}{dt} = \frac{-\frac{gt}{2}\left(2 + \frac{K}{m}\dot{x}_o t\right) + \dot{h}_o}{1 + \frac{K}{m}\dot{x}_o t} = \text{Hdot} \qquad (11)$$

$$x = \frac{m}{K}\ln\left(1 + \frac{K}{m}\dot{x}_o t\right) \qquad (12)$$

$$h = \left(\frac{\dot{h}_o}{\dot{x}_o}\frac{m}{K} + \frac{1}{2}\frac{m^2}{K^2}\frac{g}{\dot{x}_o^2}\right)\ln\left(1 + \frac{K}{m}\dot{x}_o t\right) - \frac{gt}{4}\left(\frac{2}{\dot{x}_o}\frac{m}{K} + t\right) + h_o \qquad (13)$$

$$V = \sqrt{\dot{h}^2 + \dot{x}^2} \qquad (14)$$

$$\gamma = \text{arctangent}\ \frac{\dot{h}}{\dot{x}} \qquad (15)$$

Figure 10. Further modifications on flight path equations.

$$\dot{x} = \frac{-K\dot{x}^2 g}{W}\Delta t \qquad (16)$$

$$\dot{h} = \left(\frac{-K\dot{x}\dot{h}g}{W} - g\right)\Delta t \qquad (17)$$

$$\dot{x} = \dot{x} + \Delta\dot{x} \qquad (18)$$

$$\dot{h} = \dot{h} + \Delta\dot{h} \qquad (19)$$

$$\Delta x = \dot{x}\,\Delta t \qquad (20)$$

$$\Delta h = \dot{h}\,\Delta t \qquad (21)$$

$$x = x + \Delta x \qquad (22)$$

$$h = h + \Delta h \qquad (23)$$

$$V = \sqrt{\dot{h}^2 + \dot{x}^2} \qquad (24)$$

$$\gamma = \text{arctangent}\ \frac{\dot{h}}{\dot{x}} \qquad (25)$$

Figure 11. Numerical modifications of flight path equations.

Examination of the equations in figure 10 shows that equations (12) and (13) become indeterminate if K, that is, drag equals zero. Also, equation (13) becomes indeterminate if Xdot equals zero, that is for vertical flight. Suitable error returns have been put into the program, but all possibilities have NOT been checked out.

We have modified the equations of figure 10 for numerical integration exactly as we modified those of figure 3. The results are presented in figure 11.

Program TRAJ/GD1

Program TRAJ/GD1 presents tabulated values of V, h, flight path angle (gamma), and range (X) from equations (24), (23), (25), and (22), for comparison against the values from equations (14), (13), (15), and (12). The computational sequence is exactly as in program VERTRAJ1. The user is asked to input initial values of altitude, velocity, and gamma. If a zero value is input for velocity or gamma, an error return appears and the user is then asked for the 'drag-weight parameter', C(D)*S/W, from which K will be determined. If a zero value is input another error return is touched off. In this case, however, the computation of H(AN) and X(AN) is aborted but the remainder of the calculation is continued. The reasons for these error returns were discussed in the previous section. Finally the integrating interval and the print interval are entered. A listing of program TRSJ/GD1 is given in figure 12.

```
 10 REM///////////////////////////////////////////////////////////////
 20 REM
 30 REM////////////// P r o g r a m   T R A J / G D 1 //////////////
 40 REM
 50 REM//////////// Written by  -  Alfred A. Adler, Ph.D.  ///////////
 60 REM
 70 REM/////////////// Version  1.0  -  January  1980 ///////////////
 80 REM
190 G=32.174
200 S=41.5E-6
202 R9=57.29578
205       REM ****** INPUT DATA ******
210 INPUT"Initial altitude ?",H0
220 P=.002378*EXP(-S*H0)
230 INPUT"Initial velocity ?",V0
235 IF V0=0 THEN 244
240 INPUT"Initial gamma ?",G0
242 IF G0<=89.8 THEN 255
244 !\!"***  THE ANALYTICAL EQUATION FOR H BECOMES INDETERMINATE"
245 !"          IF GAMMA=90 DG. ALSO THE DRAG ASSUMPTION IS NOT"
247 !"              GOOD FOR HIGH ANGLE TRAJECTORIES. TRY AGAIN.  ***"\!
248 IF V0=0 THEN 230
250 GOTO 240
255 INPUT"C(D)*S/W = ?",D1
260 IF D1<>0 THEN 290
270 !\!"***  THE ANALYTICAL EQUATIONS FOR X AND H BECOME INDETERMINATE"
280 !"          IF DRAG=0. H(AN) AND X(AN) WILL NOT BE PRINTED.  ***"\!
290 C=D1*P/2*G
300 INPUT"What is the integrating interval ?",T1
310 INPUT"What is the print interval ?",T2
320       REM ****** INITIALIZE ******
330 G1=G0/R9
340 G4=G0
350 X4=V0*COS(G1) \REM X4=Xdot
360 X8=X4 \REM X8=Xdot(initial)
370 H4=V0*SIN(G1) \REM H4=Hdot
380 H8=H4 \REM H8=Hdot(initial)
390 X=0\H=H0\V=V0\Z=0
400!
410 !TAB(2),"T",TAB(7),"H(NI)",TAB(15),"H(AN)",TAB(22),
420 !"V(NI)",TAB(29),"V(AN)",TAB(35),"G(NI)",TAB(41),"G(AN)",
430 !TAB(49),"X(NI)",TAB(57),"X(AN)"
435       REM ****** TRAJECTORY COMPUTATION ******
440 FOR T=0 TO 1000 STEP T1
445       REM ****** ANALYTIC ******
450 L1=1+C*X8*T
460 X7=X8/L1 \REM X7=Xdot (ANAL)
470 H7=(-G*T/2*(1+L1)+H8)/L1 \REM H7=Hdot (ANAL)
480 V9=SQRT(X7^2+H7^2) \REM V9=V (ANAL)
490 IF H7<0 THEN V9=-V9
500 IF X7<>0 THEN 520
505 IF H7>0 THEN G9=90
508 IF H7=0 THEN G9=0
510 IF H7<0 THEN G9=-90
515 GOTO 530
520 G9=ATN(H7/X7)*R9
530 L2=LOG(L1)
540 IF D1=0 THEN 570
550 X9=L2/C \REM X9=X (ANAL) & H9=H (ANAL)
560 H9=(H8/X8/C+1/2/C^2*G/(X8^2))*L2-G*T/4*(2/C/X8+T)+H0
570 IF T/T2<>INT(T/T2) THEN 630
580 IF D1=0 THEN 610
590 !%4F0,T,%8F0,H,H9,%7F0,V,V9,%6F1,G4,G9,%8F0,X,X9
600 GOTO 620
610 !%4F0,T,%8F0,H,"          ",%7F0,V,V9,%6F1,G4,G9,%8F0,X,"        "
620 IF Z=1 THEN EXIT 180
625       REM ****** NUMERIC ******
630 X2=-C*X4^2*T1 \REM X2=DELTA Xdot (N.I.)
640 H2=(-C*X4*H4-G)*T1 \REM H2=DELTA Hdot (N.I.)
650 X4=X4+X2 \REM X4=Xdot (N.I.)
660 H4=H4+H2 \REM H4=Hdot (N.I.)
670 V=SQRT(X4^2+H4^2)
680 IF H4<0 THEN V=-V
690 IF X4<>0 THEN 710
695 IF H4>0 THEN G4=90
698 IF H4=0 THEN G4=0
700 IF H4<0 THEN G4=-90
705 GOTO 720
710 G4=ATN(H4/X4)*R9
720 X3=X4*T1 \REM X3=DELTA X (N.I.)
730 H3=H4*T1 \REM H3=DELTA H (N.I.)
740 X=X+X3
750 H=H+H3
760 IF V<0 THEN IF H<0 THEN Z=1
770 NEXT
READY
```

Figure 12. Program TRAJ/GD1

It is emphasized again that this program and the one following have not been exhaustively checked for possible malfunctions. They will not, however, give wrong answers with no warning. For example, in order to get all the information desired on an 8½ by 11 inch page, the columns were crowded as much as possible. If very high values of any of the variables are attained during the trajectory, BASIC will abort the run and complain about a 'format error' in line 590 (or 610). These programs were designed to test numerical integration routines, not to compute trajectories, although within the limits of the assumptions, the results are correct if the program runs to completion with no apparent malfunctions.

Sample Runs on TRAJ/GD1

Again we repeat the run of figure 6 as closely as we can (as a check, see figure 13). The fractional differences are due to the slightly off vertical launch; otherwise everything checks. This run incidentally took 63 seconds compared to only 9 seconds for the run of figure 6. We only added a bit of complication to the equations yet the running time went up by a factor of 7. Since we are not anywhere near a set of equations that could even remotely be considered as realistic, it is apparent that large improvements must be made in the integration schemes if reasonable times are to be achieved for realistic equations.

```
Initial altitude ?500
Initial velocity ?1000
Initial gamma ?89.8
C(D)*S/W = ?0

***  THE ANALYTICAL EQUATIONS FOR X AND H BECOME INDETERMINATE
        IF DRAG=0. H(AN) AND X(AN) WILL NOT BE PRINTED.   ***

What is the integrating interval ?1
What is the print interval ?10

  T    H(NI)    H(AN)   V(NI)  V(AN) G(NI) G(AN)    X(NI)    X(AN)
  0.     500.            1000.  1000.  89.8  89.8       0.
 10.    8730.             678.   678.  89.7  89.7      35.
 20.   13743.             357.   357.  89.4  89.4      70.
 30.   15539.              35.    35.  84.3  84.3     105.
 40.   14117.            -287.  -287. -89.3 -89.3     140.
 50.    9478.            -609.  -609. -89.7 -89.7     175.
 60.    1621.            -930.  -930. -89.8 -89.8     209.
 70.   -9453.           -1252. -1252. -89.8 -89.8     244.

Initial altitude ?
STOP IN LINE 210
READY
```

Figure 13. Program TRAJ/GD1 using run of figure 6.

We now use the capability built into TRSJ/GD1. The trajectory of a vehicle launched at 45° and having a typical drag value is shown in figure 14. This run was made using an integrating interval of 1 second and took 48 seconds of machine time. However, due to the lower launch angle and the drag, the trajectory was 40 seconds long instead of 70 seconds as before. The running time was, therefore, longer per second of flight time than any previous trajectory. Note that the discrepancy between the numerical and analytical integration is significantly worse than before. A second run made with an integrating interval of .1 second is shown in figure 15. The agree-

```
RUN190

Initial altitude ?500
Initial velocity ?1000
Initial gamma ?45
C(D)*S/W = ?.0013
What is the integrating interval ?1
What is the print interval ?5

  T    H(NI)    H(AN)   V(NI)  V(AN) G(NI) G(AN)    X(NI)    X(AN)
  0.     500.     500.   1000.  1000.  45.0  45.0       0.       0.
  5.    3238.    3380.    750.   755.  36.8  37.0    3201.    3262.
 10.    4836.    5115.    574.   580.  24.5  25.0    5958.    6076.
 15.    5473.    5883.    465.   471.   7.0   8.0    8381.    8551.
 20.    5271.    5806.   -427.  -429. -14.0 -12.6   10542.   10759.
 25.    4315.    4972.   -450.  -449. -33.5 -32.1   12494.   12752.
 30.    2667.    3441.   -514.  -509. -48.1 -46.9   14273.   14569.
 35.     374.    1263.   -599.  -592. -58.1 -57.2   15908.   16238.
 40.   -2529.   -1525.   -693.  -685. -64.9 -64.3   17421.   17782.

Initial altitude ?
STOP IN LINE 210
READY
```

Figure 14. Trajectory of a vehicle launched at 45 degrees and having a typical drag value.

ment between numerical and analytical is much improved, however, this run took 480 seconds. It was considered impractical to use an interval of .01 second as was done with the run of figure 7, since the run would have taken 1 hour and 20 minutes. Obviously, obtaining reasonable accuracy is already taking an unreasonable amount of time.

```
RUN190

Initial altitude ?500
Initial velocity ?1000
Initial gamma ?45
C(D)*S/W = ?.0013
What is the integrating interval ?.1
What is the print interval ?5

  T     H(NI)    H(AN)   V(NI)   V(AN) G(NI) G(AN)    X(NI)    X(AN)
  0.     500.     500.   1000.   1000.  45.0  45.0       0.       0.
  5.    3366.    3380.    755.    755.  37.0  37.0    3256.    3262.
 10.    5087.    5115.    580.    580.  25.0  25.0    6065.    6076.
 15.    5842.    5883.    470.    471.   7.9   8.0    8534.    8551.
 20.    5753.    5806.   -429.   -429. -12.7 -12.6   10737.   10759.
 25.    4906.    4972.   -449.   -449. -32.2 -32.1   12727.   12752.
 30.    3365.    3441.   -510.   -509. -47.1 -46.9   14540.   14569.
 35.    1175.    1263.   -593.   -592. -57.3 -57.2   16205.   16238.
 40.   -1625.   -1525.   -685.   -685. -64.3 -64.3   17746.   17782.

Initial altitude ?
STOP IN LINE 210
READY
BYE
```

Figure 15. Run of Program TRAJ/GD1 with integrating interval of .1 second.

Program TRAJ/GD2

If we make essentially the same modifications to TRAJ/GD1 as we made to VERTRAJ1, we come up with program TRAJ/GD2 in analogy to VERTRAJ2. Again in the interest of economy we present only that part of the listing below REM******NUMERIC******, see figure 16. The remainder of the listing is identical to that shown in figure 12.

```
625        REM ****** NUMERIC ******
630 X2=-C*X4^2*T1 \REM X2=DELTA Xdot (N.I.)
640 H2=(-C*X4*H4-G)*T1 \REM H2=DELTA Hdot (N.I.)
650 X4=X4+X2 \REM X4=Xdot (N.I.)
660 H4=H4+H2 \REM H4=Hdot (N.I.)
662 X6=(X4+X5)/2
664 H6=(H4+H5)/2
670 V=SQRT(X4^2+H4^2)
672 REM YOU DON'T USE H6 AND X6 FOR V BECAUSE YOU WANT
674 REM   INSTANTANEOUS V NOT AN AVERAGE OVER TIME.
680 IF H4<0 THEN V=-V
690 IF X4<>0 THEN 710
695 IF H4>0 THEN G4=90
698 IF H4=0 THEN G4=0
700 IF H4<0 THEN G4=-90
705 GOTO 720
708 REM YOU DON'T USE H6 AND X6 FOR G4 BECAUSE YOU WANT
709 REM   INSTANTANEOUS G4 NOT AN AVERAGE OVER TIME.
710 G4=ATN(H4/X4)*R9
720 X3=X6*T1 \REM X3=DELTA X (N.I.)
730 H3=H6*T1 \REM H3=DELTA H (N.I.)
740 X=X+X3
750 H=H+H3
760 IF V<0 THEN IF H<0 THEN Z=1
770 NEXT
READY
BYE
```

Figure 16. Program TRAJ/GD2

Sample Runs on TRAJ/GD2

We repeat the run of figure 9 as closely as we can, as a check, see figure 17. Again the fractional differences are due to the slightly off vertical launch. This run took 70 seconds compared to 63 seconds for the run of figure 13. The additional time was of course due to the added complication of the iterative procedure in program TRAJ/GD2. The runs of figures 14 and 15 are repeated in figure 18 and 19. The running time for figure 18 was 53 seconds versus 48 for figure 14, and the running times of figures 19 and 15 were 10 times as long. The differences between the numerical and the analytical results are about one-third as much as in figures 18 and

19 as they are in figures 14 and 15, whereas the running times only differ by about 10%. Once again we see that averaging the rate data over the integration interval yields a large improvement in accuracy with only a small increase in running time. We observed the same effect before in program VERTRAJ2.

```
Initial altitude ?500
Initial velocity ?1000
Initial gamma ?89.8
C(D)*S/W = ?0

***  THE ANALYTICAL EQUATIONS FOR X AND H BECOME INDETERMINATE
          IF DRAG=0. H(AN) AND X(AN) WILL NOT BE PRINTED.  ***

What is the integrating interval ?1
What is the print interval ?10

  T    H(NI)   H(AN)  V(NI)  V(AN) G(NI) G(AN)   X(NI)   X(AN)
  0.    500.          1000.  1000.  89.8  89.8     0.
 10.   8891.           678.   678.  89.7  89.7    35.
 20.  14065.           357.   357.  89.4  89.4    70.
 30.  16022.            35.    35.  84.3  84.3   105.
 40.  14761.          -287.  -287. -89.3 -89.3   140.
 50.  10282.          -609.  -609. -89.7 -89.7   175.
 60.   2586.          -930.  -930. -89.8 -89.8   209.
 70.  -8327.         -1252. -1252. -89.8 -89.8   244.

Initial altitude ?
STOP IN LINE 210
```

Figure 17. Program TRAJ/GD2 repeating the run of figure 9.

```
RUN190

Initial altitude ?500
Initial velocity ?1000
Initial gamma ?45
C(D)*S/W = ?.0013
What is the integrating interval ?1
What is the print interval ?5

  T    H(NI)   H(AN)  V(NI)  V(AN) G(NI) G(AN)   X(NI)   X(AN)
  0.    500.    500.  1000.  1000.  45.0  45.0      0.      0.
  5.   3367.   3380.   750.   755.  36.8  37.0   3254.   3262.
 10.   5070.   5115.   574.   580.  24.5  25.0   6051.   6076.
 15.   5798.   5883.   465.   471.   7.0   8.0   8503.   8551.
 20.   5676.   5806.  -427.  -429. -14.0 -12.6  10689.  10759.
 25.   4793.   4972.  -450.  -449. -33.5 -32.1  12660.  12752.
 30.   3212.   3441.  -514.  -509. -48.1 -46.9  14455.  14569.
 35.    981.   1263.  -599.  -592. -58.1 -57.2  16103.  16238.
 40.  -1861.  -1525.  -693.  -685. -64.9 -64.3  17627.  17782.

Initial altitude ?
STOP IN LINE 210
```

Figure 18. Program TRAJ/GD2 repeating the run of figure 14.

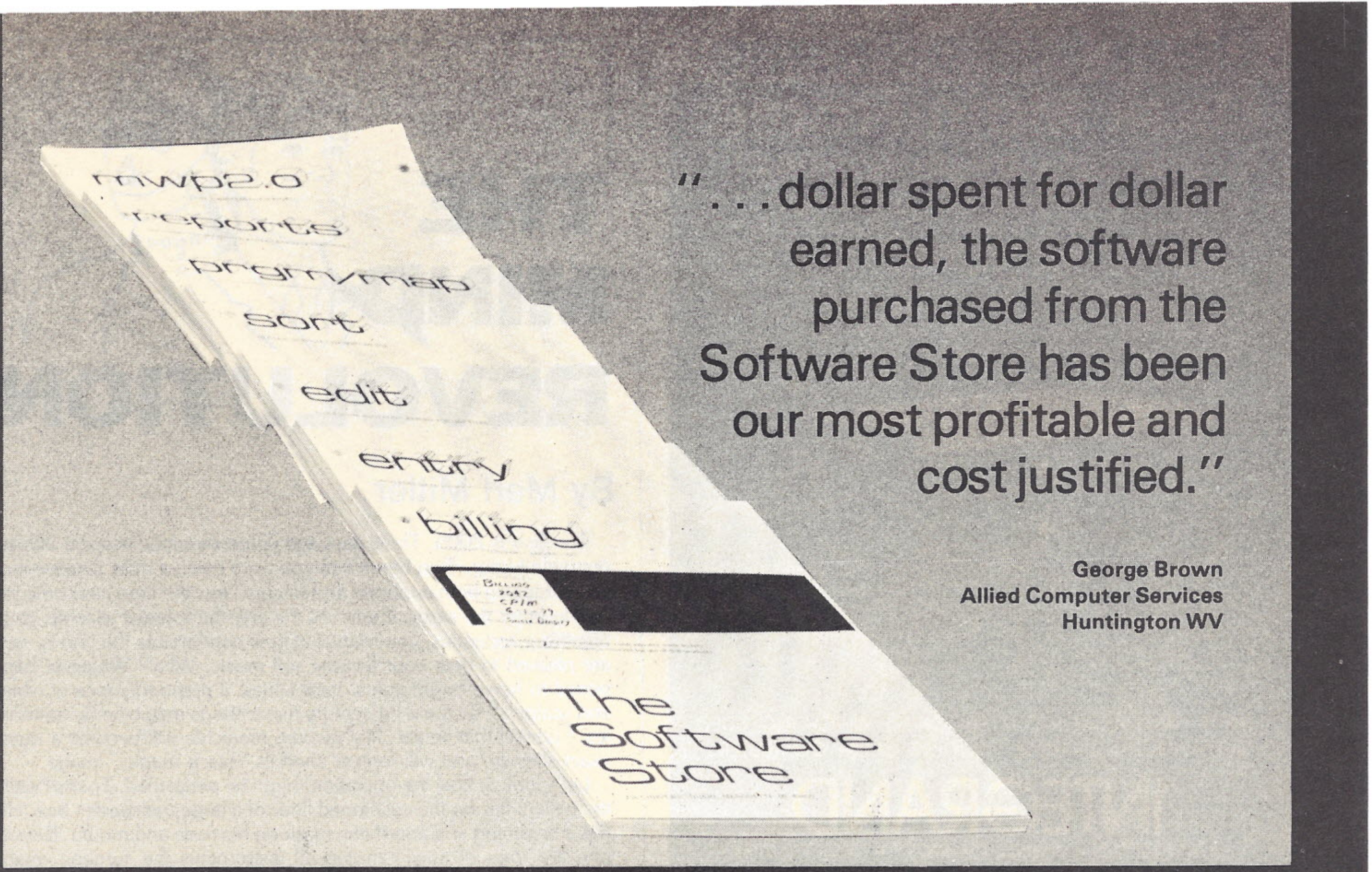
```
RUN190

Initial altitude ?500
Initial velocity ?1000
Initial gamma ?45
C(D)*S/W = ?.0013
What is the integrating interval ?.1
What is the print interval ?5

  T    H(NI)   H(AN)  V(NI)  V(AN) G(NI) G(AN)   X(NI)   X(AN)
  0.    500.    500.  1000.  1000.  45.0  45.0      0.      0.
  5.   3379.   3380.   755.   755.  37.0  37.0   3261.   3262.
 10.   5110.   5115.   580.   580.  25.0  25.0   6074.   6076.
 15.   5874.   5883.   470.   471.   7.9   8.0   8546.   8551.
 20.   5794.   5806.  -429.  -429. -12.7 -12.6  10752.  10759.
 25.   4954.   4972.  -449.  -449. -32.2 -32.1  12743.  12752.
 30.   3419.   3441.  -510.  -509. -47.1 -46.9  14558.  14569.
 35.   1235.   1263.  -593.  -592. -57.3 -57.2  16225.  16238.
 40.  -1558.  -1525.  -685.  -685. -64.3 -64.3  17766.  17782.

Initial altitude ?
STOP IN LINE 210
```

Figure 19. Program TRAJ/GD2 repeating the run of figure 15.

FURTHER IMPROVEMENTS

Unfortunately, closer approaches to reality in the equations increase the running time so rapidly that trajectory equations must be numerically integrated using techniques very much more sophisticated than those discussed herein. Reference back to figure 2 and a reminder of the simplifying assumptions included in their derivation should convince the reader.

There are, however, many equations in engineering and the sciences which require numerical integration and for which the methods presented herein are quite adequate. Hopefully the reader has gained an appreciation of the difficulties, an awareness of some of the pitfalls, and a useful tool for the solution of some of the more intractable differential equations.□



Why reinvent the wheel? The Software Store supplies complete program systems written in easy to use Microsoft BASIC for Radio Shack Model II, Altair/MITS, TEI, Cromemco, North Star, Processor Tech, Altos, Ohio Scientific, Billings, IMSAI, Digital Micro Systems and other Z80 & 8080 based computers. Our growing family of products is divided into three categories: *application utilities, systems* and *system utilities*.

The *application utilities* are the basic building blocks for application program systems. Almost every application can be made of a key-to-disk data entry segment, a file edit segment, a sort/merge segment, a record selection segment and a report & file update segment. These functions are carried out by the ENTRY, EDIT, SORT, SELECT and REPORTS systems, respectively. Application utilities consist of two programs: one for interactive task definition and the other for task execution. Once defined, a task may be executed any number of times or easily revised.

Application utilities permit rapid solutions to satisfy each user's unique requirements. Many first time computer users have built respectable application systems using our utilities and self instructive documentation. Computer stores and consultants utilize our products to generate custom systems for their clients. Because of the flexible and interactive design of the task definition programs, previously defined systems can be easily revised to meet changing needs.

The *systems* are complete packages for a specific application. Systems are fabricated from application utilities together with application specific programs. For example, our Accounts Receivable System utilizes the ENTRY, EDIT, SORT, SELECT and MWP systems along with six special billing system programs.

The MWP system is a complete word processing system with flexible user defined "name & address" files. The "name and address" information and date can be inserted throughout a document. The documents might be reports, manuals, mailing labels, letters or legal documents.

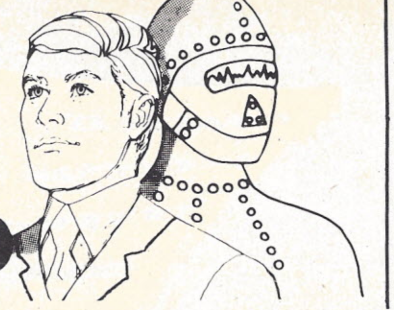
The *system utilities* include programming tools such as the Program Map BASIC cross reference program along with general utilities such as the Disk Fix file recovery program, the Disk Copy (1D & 2D) diskette copy program, the TX-RX file transfer and media conversion programs and the CATALOG diskette library index program.

To find out more about our growing family of software products, contact your local computer dealer for a demonstration or contact us.

The Software Store

706 Chippewa Square
Marquette MI 49855
(906) 228-7622

THE MIND REVOLUTION



By Merl Miller

The year is 2005. Three days ago you entered the hospital with an operable tumor. Two hours ago you were given a mild sedative that has made you both euphoric and sleepy. Now the time has come for you to enter the surgical unit. As the anesthesiologist attaches your breathing apparatus, you marvel at how comfortable the unit is, and are pleased to hear your favorite soft music. What? Whatever happened to harsh bright lights, hard tables, a preponderance of white and scalpels? By the 21st century these things may be only memories. I predict that by the 21st century medicine will become a more exact science, and will forever shed its "black magic" image.

Let's look at how the operation might be performed. To start with, the patient lies on the cushioned floor of a large transparent box. He has a breathing and anesthetic mask on his nose and mouth. Beside him are rows of trays carrying all instruments the surgeon might need. Each instrument's handle is a hollow tube. Scissors, forceps, sewing devices. . .all have threaded ends that can easily be attached to the surgeon's instrument-holding rods. The primary surgical instrument is the laser.

Another sterilized instrument tray stands ready to replace the first through a sterile lock. The box in which the patient lies is sterilized between operations with steam and ultraviolet rays. The roof of the box carries a television camera. The surgeon can move the camera lens anywhere within the box. Back-up cameras are strategically located throughout the box. Each camera is capable of from two-to-ten times magnification, if needed.

The box lid has ten arms, each of which ends in a rod that can fit any of the instruments, including the laser. The surgeon sits at a control desk facing the television screens. The control desk is attached to the surgical table, and all interconnecting rods run between. If the surgeon wants, he can look over the control desk directly onto the operating table. He inserts one of his arms into a close fitting electrosensitive glove that reaches from fingertips to shoulder. These gloves are crucial to the entire procedure so I will explain how they work separately. First, let's see how the surgeon uses them.

Each fingertip of the glove is connected to the computer, which is in turn connected to the rods. The surgeon uses one hand to control the equipment, the other hand to operate. The electrosensitive glove emulates the surgeon's hand movements exactly. If the surgeon makes a sewing movement, it causes the rod attached to the needle to sew. If the surgeon wants to make an incision, he can do so using his index finger. All he has to do is tell the laser that it is operated by the index finger. Then, whatever the surgeon does with his index finger, will be duplicated by the laser.

The surgeon can manipulate his instruments with as much freedom and dexterity as if they were directly in his hands. But the control system gives him a steadiness of hand far greater than could otherwise be achieved. Not only do the holders follow his finger exactly, but he can feel the resistance to movement and the weight of the tools as if he were holding them. If he wants to hold an instrument in an exact position, he has only to turn it off. The instruments can be of wide variety in size, so he can limit the size of the incision necessary to reach difficult areas.

The main television camera can be controlled by head movements so that, when the surgeon looks at something, so does the camera. The side cameras are always on, so he can check them at will. He can adjust the lens to give the exact picture needed.

A little farfetched? Not really. Some of these things are in use now. For instance, a camera that looks where you do has been used in military applications for some time.

Let's now turn our attention to the most important piece of equipment — the electrosensitive gloves. They will be lined with electrodes and have a few microprocessors and other integrated circuits embedded in them. These devices will be used for only one thing: producing output that can be interpreted by the computer. Each surgeon will have his own glove, and his own interpretation module, "trained" to respond to signals from his arm.

Each movement of each finger will be interpreted by a group of electrodes. As the electrodes sense movement, a signal will be sent to an internal microprocessor (or glove) where a movement signal is created. This signal is sent to the interpreter and from there to the main computer for action. As computers operate in picoseconds, and humans still operate in seconds, the system should be extremely sensitive.

This leads us to my final prediction. If you remember, last month's column had a short comment about direct "brain link" communication with a computer. I foresee the time when you will be able to operate a variety of devices simply by thinking about them. Such an application in surgery is fascinating.

Imagine a situation where a surgeon has been trained to have a certain physical feel for operating. During his schooling he has an opportunity to practice his skills using both computer simulation and some of the methods in use today. He starts his internship by watching other surgeons at work and assisting in minor surgery. All minor surgery is performed in the manner previously described. Eventually, he is allowed to participate in minor surgery until he exhibits a prescribed degree of proficiency. At this point, he takes his last series of medical exams, and is awarded a degree in surgery. He is now allowed to perform minor surgery and diagnose problems for referral to a master surgeon.

I foresee the time when you will be able to operate a variety of devices simply by thinking about them. Such an application in surgery is fascinating.

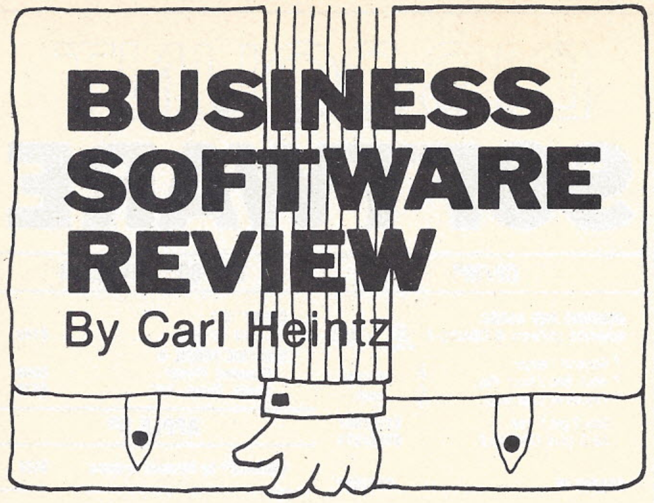
At some point, he may decide to become a master surgeon. Again, there will be a series of time and proficiency requirements to meet, and he will have to specialize in a particular field. This last phase of training will be the most difficult because it involves a great deal of mental discipline. However, most people who begin the course will probably complete it. Remember, by definition we are dealing with a group of overly intelligent people, starting with phase one, who find this training both exhilarating and fascinating.

It seems to me that at least some of the training involves total isolation. It won't do for the surgeon to be distracted when making an incision. He must learn to concentrate precisely on what he is doing.

Let's look in on a 21st century master surgeon at work. He sits at a control desk surrounded by TV screens. He wears a skullcap device connected to the computer. He carefully looks at the main camera console which is connected to his skullcap so it moves when he does. The patient is bathed in pale blue light so the brighter lights of the instruments show each phase of the operation with great precision.

The small white penlight on the laser indicates exactly where the beam will fall when it is turned on. He wants to check the path of the incision so he thinks to himself, "penlight on; at my direction, hover over the patient and follow this path. Starting here, make an incision one centimeter deep and four centimeters long. Record this for the laser." Addressing the laser, he thinks, "laser, start at the specified point, and make the recorded incision."

This sounds terribly slow, but remember he is thinking. The entire operation can be done at a modified speed of thought. It will require an enormous amount of discipline, practice and training; but it can be done.☐



Selector III C-2, produced by Micro Ap in San Ramon, CA advertises heavily as an "information management system." Promotional literature stresses a distinction between the accounting function and the "storing, processing and reporting of information," and it is these latter functions that Selector is designed to perform.

In substance, what Selector does is give the user a multi-key "indexed" access to files which the user defines. Through this ISAM-type of file management, the user can input, sort, extract and manipulate data, producing custom reports to specific user needs. As such, this puts Selector in the class of a practical "data base management system."

Selector runs under the CP/M operating system, using either CBASIC or M BASIC. It needs at least 25K of memory space, which means that with CP/M, plan on at least a 48K system. It is designed for an 8080, 8085 or Z-80 microprocessor and two disk drives (while it is possible to get by with one, it's very difficult and time-consuming). Disk formats supported include those found on Dynabyte, North Star, Micropolis (Vector MZ), TRS-80, Cromemco and others. The system costs about $350, including a diskette and 50-page manual.

A word of warning — Selector is not designed for the novice. Since sophistication and power have the price of complexity, this program should not be a first purchase. However, once a microcomputer user has a handle on the power of CP/M and understands how to use CBASIC, there should be no hesitation.

Selector is distributed in source code, which means that changes can be made to the program before they are "compiled" and used. This also means that revisions are easy to incorporate. Micro Ap is one of the few distributors of software that takes the responsibility of "updates" seriously.

Before launching into programming with Selector, the user should spend a little time reviewing what a database is, and how such things are usually handled on micros. Since the system comes in source code and has tremendous flexibility, some work on the user end will be necessary for implementation.

Selector programs are all menu driven and have the following general divisions:

A. DEFINE — a set of programs used to create, delete, change or modify files. Included are utilities to get files from disks, and put them back (rather than having to use PIP).

B. SET — Selector operates under the program philosophy that a user loads all the information into the database and then "selects" information using the parts of the database that are needed. The SET series allow the user to define what is to be selected and what order they are to be selected in.

C. SELECT — Once a user has defined the commands to select data, the Select programs use these criteria to generate a set of file pointers — a set of directives created as the result of the programs. These pointers are, in essence, indexing schema or an ordering list used to place the files in logical order for the user's application.

D. REPORT — A series of programs which produce the desired output from the database. A command set is created determining which records shall be included, what their order shall be and what format they shall be printed in. This order can be saved for continued use, and everything from mailing labels to complex reports can be generated.

E. UPDATE — A series of programs that provides the information management. A menu allows the user to select whatever activity is desired, including making new entries, retrieving information (including a "dump" of the database), a sequential record recall, changing information, deletion of records, etc.

The user can create files with predefined sorting and report criteria for later (and continued) use, or the programs can be used to produce reports on a one-shot basis. It is possible to link other programs into the Selector input and output cycles so the user can create production programs to enter large amounts of data.

As an extra bonus, the Selector package contains applications programs including sales, inventory, accounts receivable and payable, disbursements, client records, and appointments. The purpose of all these ready-to-run programs is to show that the applications contained within Selector can go beyond the traditional database concept.

The experienced programmer will have no difficulty using Selector as a "nested" utility to generate some elegant applications programs. For the programmer with a limited background, the examples and the source code are more than sufficient.

DEFINING THE DATABASE

The Selector programs used to define files are completely menu driven. It is relatively easy to set up new files. The user is asked for the filename, the number of fields and the program sets up a little table that the user completes in order to define data elements. There are six data types which are supported:

Alphanumeric	Numbers to 999.999 +/−
Alphanumeric key (1st 10 characters)	Numbers to 999,999.999 +/−
Numeric key (up to 999,999.999 +/−)	Numbers to 999,999,999.999 +/−

The most obvious question from a novice user is "why so many types of data?" The answer is the effect upon memory requirements and the corresponding limitations on the systems as to the number of transactions that can be accommodated in a session.

Note that decimals of up to three places are supported — which is more than adequate for any business application. Additionally, the program contains edit capabilities. The nature of a data element is "built-into" the file so that report writing is simpler. Types of editing include:

Dates (as either mmddyy or yymmdd, displayed with hyphens, such as 03-15-80)
Area code (nnn)
Phone number (nnn-nnnn)
Social security number (nnn-nn-nnnn)
Decimal point precision from 0 to 3
Currency ($NNNN.NN)

After the size, in terms of number of characters and data type, has been determined by the user, Selector will compute the actual record size and display this information. Thus the user has an opportunity to know the size of the packed records before implementing it. The programmers at Micro Ap were apparently aware of a user's typical reaction to large file size. By providing the actual size and giving the user the opportunity to change the record before it is "set in concrete," the program does a great service to all overly zealous programmers.

Selector's manual contains an ominous warning about record sizes: "Bear in mind that if the actual record size being read or put is not equal to the assigned record size, the program will 'bomb.' Files that are open at this time will probably be corrupted. . ." This is more a function of CBASIC than of Selector, however.

Another limitation imposed upon Selector by CBASIC is the maximum record size. Since CBASIC limits all strings to 255 bytes, the maximum length of a record is 255 bytes.

Previously, the function of "select and sort sets" was touched upon. Selector uses these files to do the actual record selection and construction of the pointer list to the selected records.

The "select set" is created in a fashion analogous to the selection of record parameters. Again, the program is entirely menu driven, so the operator need only answer questions as they appear. Promotional literature advises that "multiple Boolean selections" can be made. For the non-mathematician user, this has less than crystal-clear meaning. What it encompasses, however, is a selection methodology that allows any combination of the following tests to be used to select a record:

Equal to	Not equal to	Less than
Between two values	In a list of	Greater than
Contained in a field	Less than or equal to	Greater than or equal to

As many as 24 criteria can be entered, and can have multiple criteria for any field. The tests can operate as test 1 and test 2 or, alternately, test 1 or test 2. That's where the Boolean reference comes in.

Once a record test is constructed, it is used to create a list of those records that pass the tests. This list, essentially an index, is called a select set by Selector. There is room for about 125 record pointers for every 1000 bytes of free memory space. In a 48K system, this allows about 2250 records selected from the file.

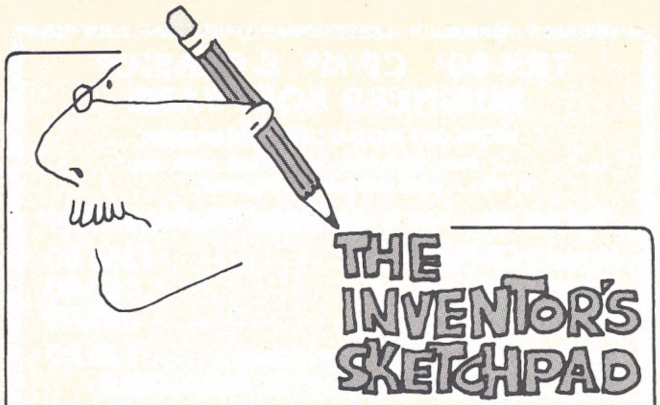
In use, the select set tests the data file sequentially, testing each field designated as a test field against the test value according to preset criteria. If the tests match, a pointer to that file is placed in an array.

REPORT

The report section of Selector is superior to most database management systems. The program allows the report to be generated in a number of sequences, including report sequence, ascending or descending key sequence using any key field or in the order specified by a "select set." "Report" also has an option to have the report include summaries, sub-summaries and even the capability to prepare minima, maxima, averages and of course grand totals. To top it off, the report program numbers the pages.

A common misconception concerning report generators for database management systems is that they allow the user to construct any kind of report. Of course this is fallacious, since a report is generally limited to a list of elements of the database, with totals and subtotals. Elaborations of the report function require custom software to be written. With Selector it is feasible, since the user has access to the file structures, and can list the "report" program to examine parts of it. Through thoughtful planning, the informational reports generated by Selector should be sufficient.

One feature of CCA's data management system missing from Selector is the ability to define one field as a computational result of another field. In other words, field 2 might be defined as field 1 x 15%. This feature is useful in some database applications; however, with proper programming of the input programs, the necessity for this kind of "internal computation" is largely minimized.☐



By Roger Garrett
Contributing Editor

A CRIME FIGHTING SYSTEM

Before I got into the computer field, I spent about a year and a half as an investigative assistant with the New Haven, Connecticut police department. One of the other aides was developing a system to assist in the apprehension of bank robbers. Essentially, she took a map of the roads around each bank and marked off the roads most likely to be used by a suspect when fleeing from the scene of the crime. She then identified those intersections where the positioning of police officers or roadblocks would probably be most effective in capturing the suspect.

The system was effective as far as it went; but it had many faults. Any change in street status (such as changing from one-way to two-way or road construction obstructing passage) rendered her maps obsolete. Whenever a bank opened a new branch, someone had to draw up a new map. Her set of maps only handled bank robberies; if someone decided to hold up a jewelry store, the maps were useless. What was really needed was a real-time interactive computer-based suspect apprehension system.

Let's assume we have a computer with a large database system and a color graphics terminal with light pen and keyboard. The database includes two forms of information. The first form comprises a graphic representation of the city streets that can be displayed and moved around on the color monitor. The other form is a set of data describing those streets: traffic direction (one-way, two-way, dead end), street linkages (how they intersect), traffic flow (approximate speed based upon time of day, day of week, and season of the year), and traffic control (where stop signs, traffic signals, etc. are and how they operate).

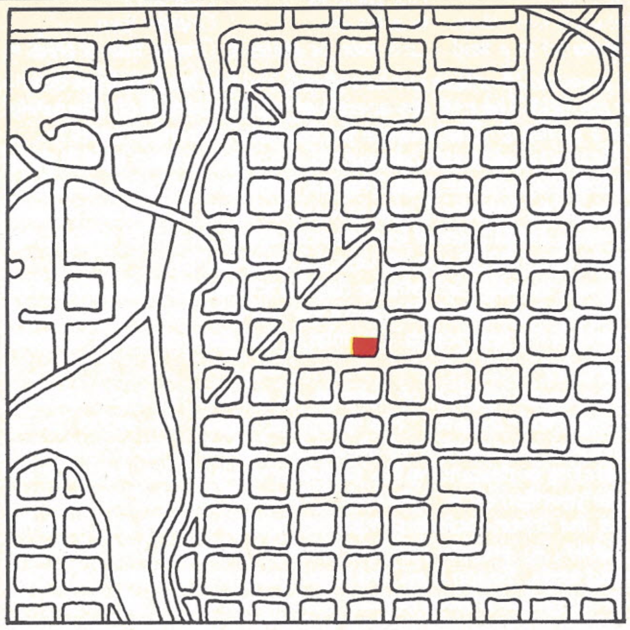
The light pen and keyboard allow the dispatcher to enter data into the system, which responds by producing appropriate displays. He can, for example, indicate to the system that a crime has occurred at some point in the city, (any point, not just at banks) and the system will display that area of the city. With special color codes being displayed in real-time, the computer assists the dispatcher in deploying police personnel to apprehend the suspects. (See figures 1 through 5.) With such a system, the computer no longer remains just a tool but becomes a partner, actually helping the operator with intelligent suggestions.

FIGURE ONE

The dispatcher receives a call that a crime has occurred at a jewelry store on the corner of Third Avenue and Fifth Street and that the suspects were seen leaving the scene of the crime in a northbound vehicle. The dispatcher picks up his light pen and touches the reset button followed by the vehicle north and crime location buttons. If he can visually locate the appropriate position on the displayed map, he touches his pen to that point. Otherwise he types in the crime location on his keyboard. The system responds by displaying a red square at the location on the map.

FIGURE TWO

The system plots the possible routes that the suspects may take. The yellow area indicates the areas that could have been reached had the suspects been running rather than riding in a car. The green sections indicate where a vehicle could have traveled in the elapsed time. The blue area indicates a 1-minute projection into the future



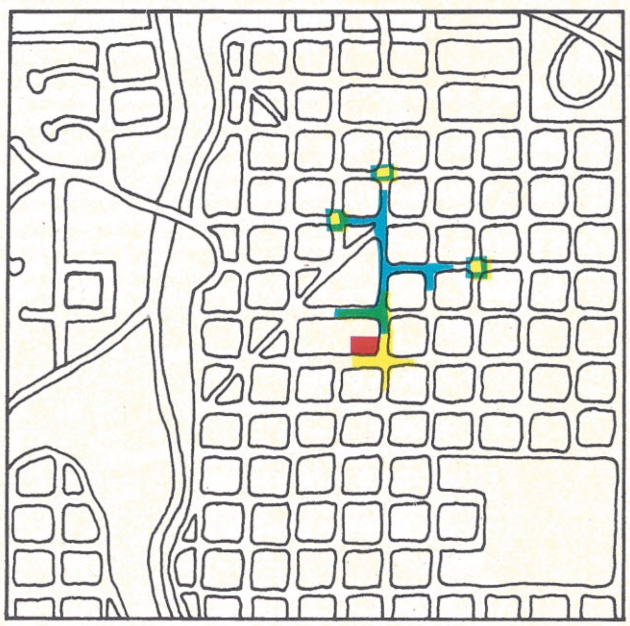
CRIME LOCATION: **Third Ave & Fifth St**
INTERCEPTION POINTS:

FRIDAY JAN 6 80
8:56:03 A.M.

N W E S

CRIME LOCATION	PATROL CAR LOCATION	OFFICER LOCATION	ROAD-BLOCK LOCATION	VEHICLE	ON FOOT	ADD	DELETE	APPRE-HENDED	RESET

Figure 1.



CRIME LOCATION: **Third Ave & Fifth St**
INTERCEPTION POINTS: **Fifth Ave & Seventh St**
Sixth Ave & Second St
Seventh Ave & Fifth St

FRIDAY JAN 6 80
8:57:42 A.M.

N W E S

CRIME LOCATION	PATROL CAR LOCATION	OFFICER LOCATION	ROAD-BLOCK LOCATION	VEHICLE	ON FOOT	ADD	DELETE	APPRE-HENDED	RESET

Figure 2.

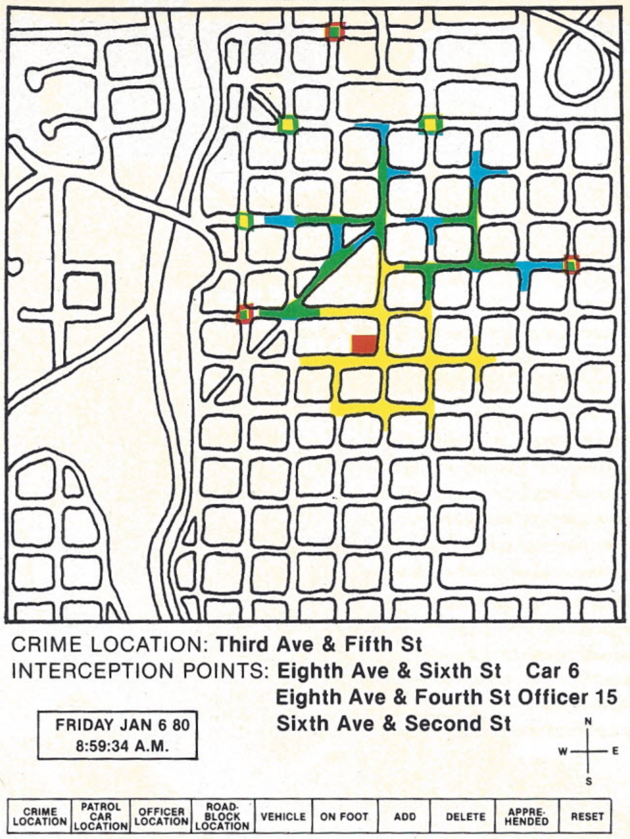


Figure 3.

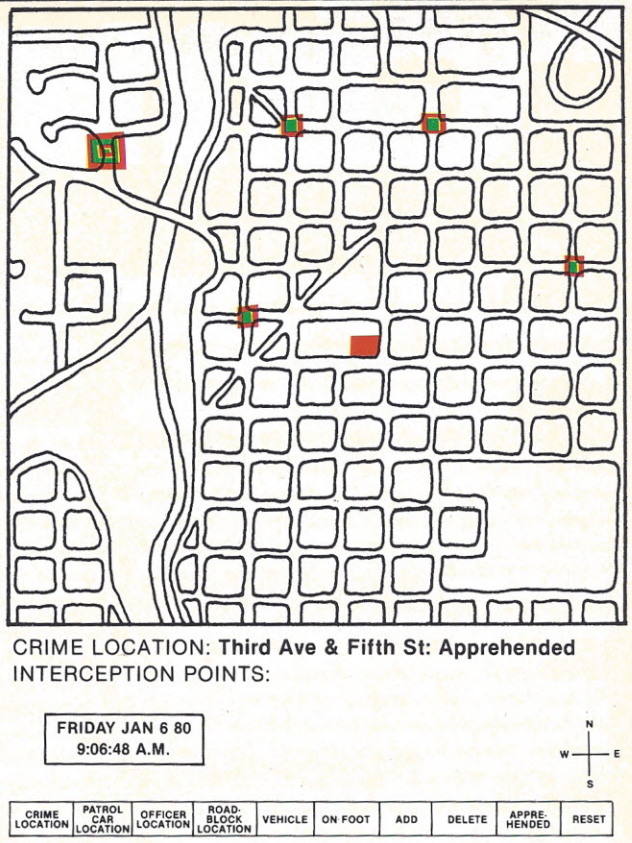


Figure 5.

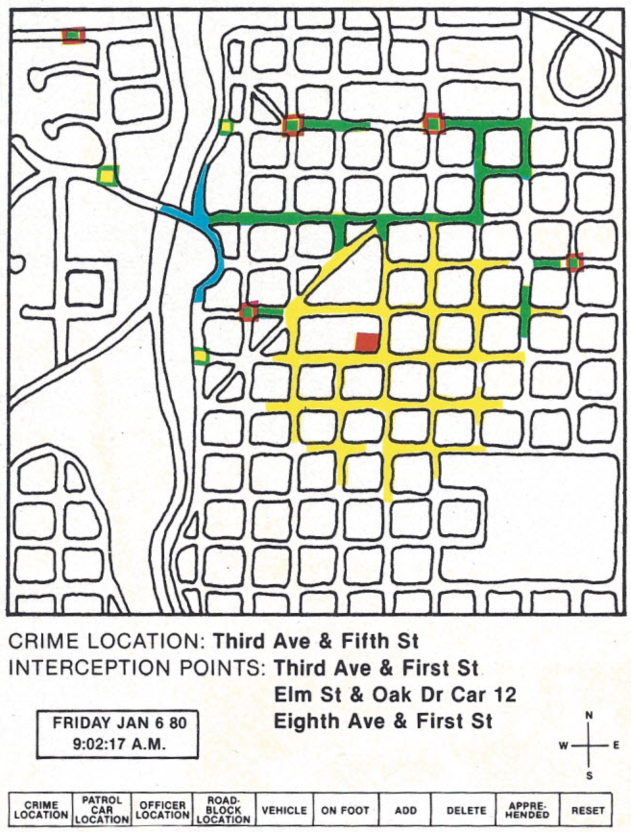


Figure 4.

where the suspects' vehicle could travel. In determining the green and blue areas, the computer takes into account that the vehicle was traveling north along Fifth Street when last seen, that certain roads are one-way streets, and also the approximate traffic flow rates for the time of day, day of week, and season of the year.

The system has also displayed several yellow-centered green squares as potential interception points and has noted their locations at the bottom of the screen. These represent intersections which the computer has determined would be best for positioning police personnel in order to apprehend the suspects.

FIGURE THREE

A few more moments have passed and the system continues to update the display. The recently displayed green-centered red squares indicate positions of police personnel who have called in and reported their positions, which the dispatcher then enters as data either via the keyboard or the light pen. As the computer updates the suggested interception points, it can now take into consideration the estimated amount of time necessary for an officer to get to that location. It also displays the police car or officer which it suggests be sent to the interception points. The dispatcher can accept the suggestions and appropriately deploy the personnel or make his own decisions. Of course, as a given police department gains confidence in the system, it might allow the system itself to do the dispatching via voice synthesis over the police radio.

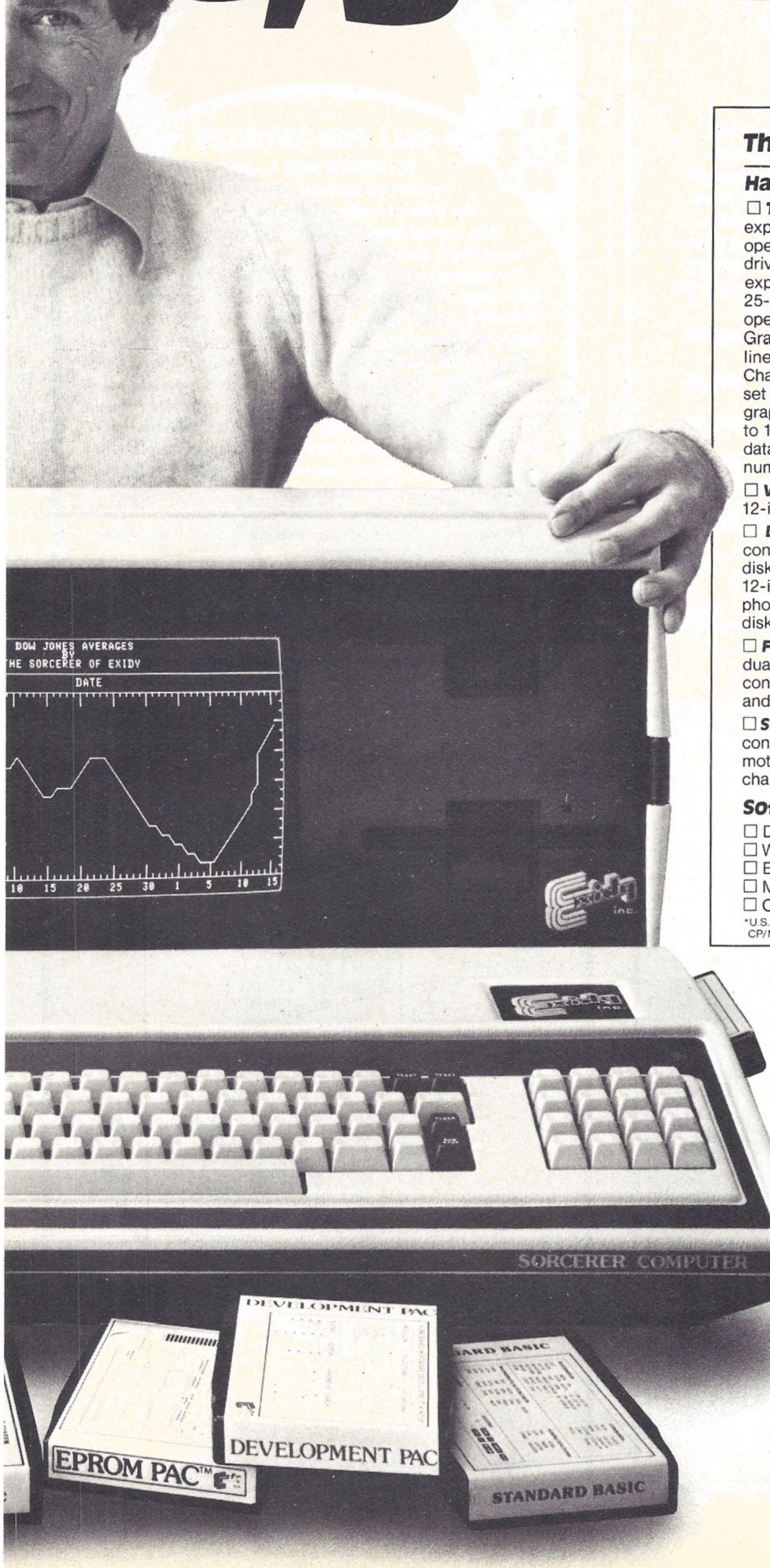
FIGURE FOUR

Several officers are dispatched and arrive at crucial intersections, effectively blocking escape routes. If the computer has access to the city's traffic light system, it might even halt traffic in some areas in order to slow down or stop the suspects. Note that there is relatively little blue on the map now, indicating a good chance of apprehension.

FIGURE FIVE

Finally, after dispatching car 12 towards the corner of Elm and Oak, the officer reports that the suspects have been apprehended, indicated by a clearing from the display of all symbols except the crime location, apprehension symbol, and police personnel symbols. When the dispatcher touches his light pen to the reset button, the symbols disappear and the system is ready for the next incident.□

me, gimme."



The best of Sorcerer

Hardware	Price*
☐ ***The Sorcerer Computer:*** 16K RAM expandable to 48K. 4K ROM-resident operating system has built-in printer driver. 50-pin I/O connector for S-100 expansion. 25-pin parallel interface; 25-pin serial interface. Serial interface operates one or two cassette recorders. Graphic resolution of 240x512, 30 lines of 64 characters, 8x8-dot matrix. Character generator contains full ASC II set (upper & lower case), plus standard graphic symbols. User may define up to 128 characters. Keyboard is 63-key data processing type, plus 16-key numeric pad.	***$1295*** (16K model) ***$1395*** (32K model) ***$1495*** (48K model)
☐ ***Video Display:*** industrial grade 12-inch CRT (P31 phosphor).	***$ 499***
☐ ***Display Disk:*** single unit package containing two quad density dual floppy disks (630K bytes) with controller and 12-inch industrial grade CRT (P31 phosphor). Includes CP/M™ and Microsoft disk-extended BASIC.	***$2995***
☐ ***Floppy Disk Subsystems:*** single dual-density floppy disk (120K bytes) and controller in one package. Includes CP/M™ and Microsoft disk-extended BASIC.	***$1150***
☐ ***S-100 Expansion Unit:*** self-contained 10-amp power supply and motherboard with 6 card slots. Daisy chain units for more slots.	***$ 419***

Software

☐ Development PAC™	***$ 99***
☐ Word Processing PAC™	***$ 199***
☐ EPROM PAC™	***$ 49***
☐ Microsoft Disk-Extended BASIC	***$ 300***
☐ CP/M™ Operating System	***$ 145***

*U.S. domestic prices only. Subject to change without notice
CP/M is a Registered Trademark of Digital Research

Satisfy that urge

When you discover the many exciting ways Sorcerer can satisfy all your "Gimme" urges, you'll say: "I'm in sheer Exidy." For the name of your nearest dealer, or additional information, write or call our Marketing Communications Department.

CIRCLE INQUIRY NO. 29

Exidy
Data Systems Division
390 Java Drive
Sunnyvale, CA 94086
(408) 734-9410

CALL FOR ARTICLES

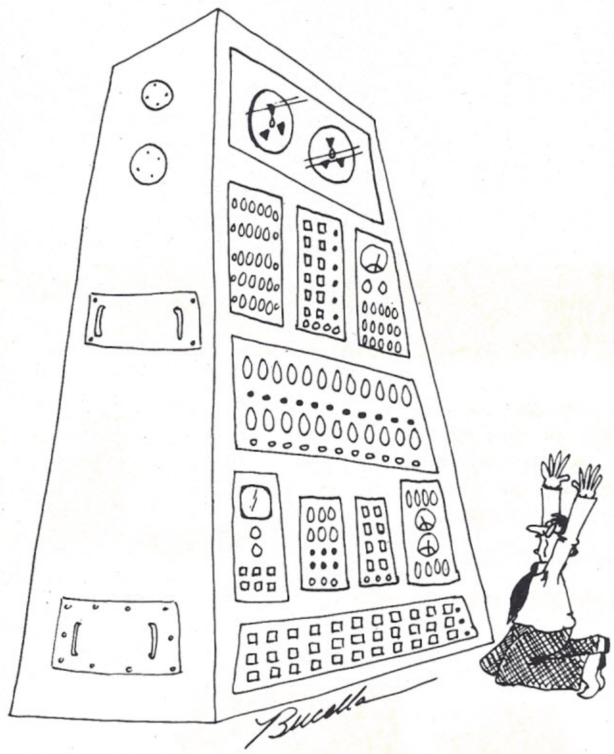
INTERFACE AGE is seeking articles on hardware and software in the areas of medical and educational application, special functions of microprocessors, telecommunications and video graphics, both screen and hardcopy.

The payment rate for articles ranges from $20 to $50 per published page. Pieces describing company projects or products will carry the company byline, but no payment is offered. Submittals should include an abstract, outline and stamped return envelope.

Manuscripts should be typed, double spaced with one-inch margins. Minimum length is four pages, unless programs are included. Photos should be numbered and have a brief description attached. Tables, listings, etc. should be on separate pages. Computer listings should be printed using a new ribbon to assure good reproduction. Authors are requested to submit a statement of their background and expertise.

The publisher assumes no responsibility for artwork, photos or manuscripts. No acknowledgement is made unless accompanied with a stamped return envelope.

For article submission or more information, write Editor, INTERFACE AGE Magazine, 16704 Marquardt Avenue, Cerritos, CA 90701. Please do not phone.



"Oh, Mighty One, that I designed. . ."

Teach Your PET to Read "Marked Cards"

By Marvin Mallon

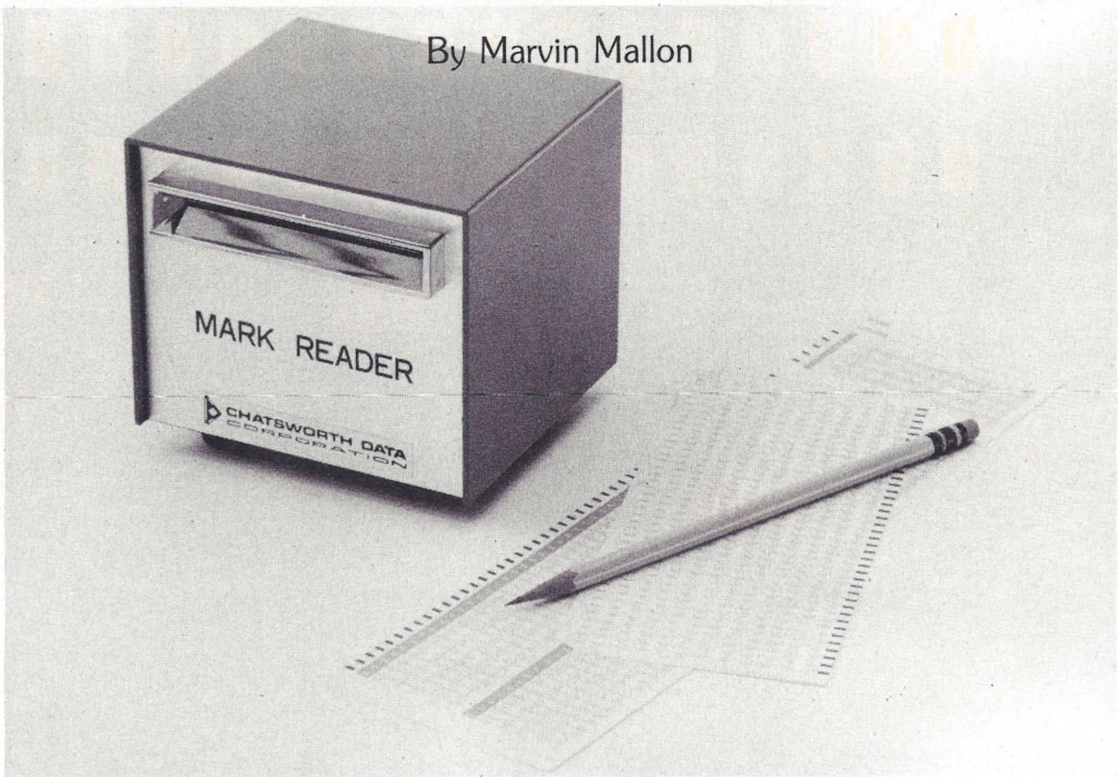


Photo courtesy of Chatsworth Data

"Marked cards" herein do not refer to the kind that may net you a fortune in Las Vegas. They refer to those ubiquitous IBM-style cards that are designed for alphanumerical data entry tasks. More specifically, the mark sense cards that are read by any device sensitive to the presence of a pencil mark. This is not to be confused with the more familiar punched cards.

For many decades, the mark sense card has been used in conjunction with mini and maxi computers for the recording of inventory, survey results and school examinations (see figure 1). It has proven to be a simple and reliable means of entering great quantities of pre-recorded data into a computer in a fast and efficient manner. With the recent introduction of the inexpensive MR500 Mark Reader manufactured by Chatsworth Data (photo), the mark sense card is now a viable consideration for the micro. Prior to the production of this small $750 unit, similar devices normally sold for many thousands of dollars and overshadowed the microcomputer both in price and size.

A key to the reasonable selling price lies in the manner in which this system "senses" or "reads" the cards. Older models used optical scanning techniques with costly electronics. This unit scans the width of the card as it passes through the reader with a series of 13 sets of metal brushes. When a pencil mark (preferably #2 or softer) comes under a brush set, the conductivity of the graphite is sufficient to provide a pulse from that channel. Twelve of the channels fall in line with the conventional spacing of a Hollerith encoded IBM card. The remaining channel (to the far left) is reserved for the sensing of pre-printed conductive timing marks which are a necessary clue as to which row is being read.

Another cost-cutting measure incorporated in the MR500 is the absence of a hopper or bulky collector box for the cards. This small, lightweight unit (4 pounds, 4½ inches cubed) is intended for a one-at-a-time card reading. Operation is simple. The card (suitably pencil marked) is pushed

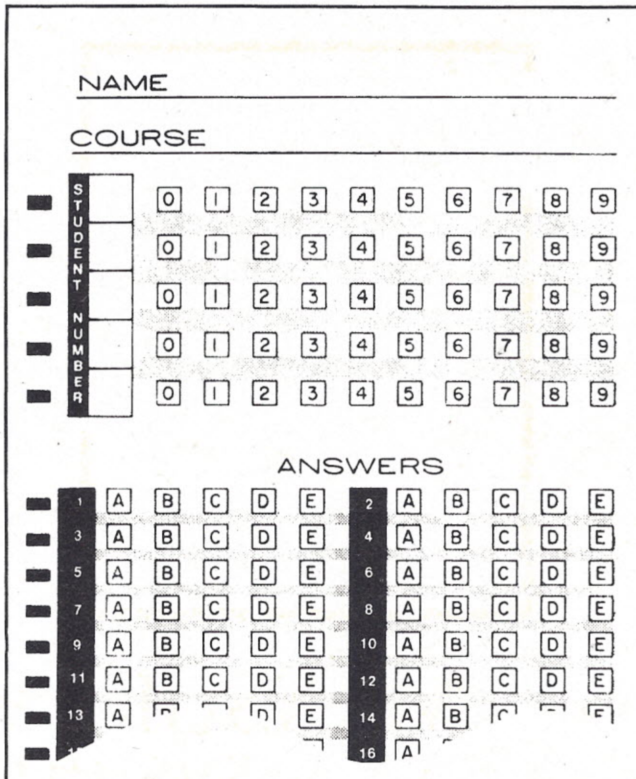


Figure 1. Special card used in the Test Scoring program. Note conductive timing marks along left edge.

into the slit-like opening at the front of the unit. A switch senses this action and a motor starts the rollers which pull the card through. It emerges directly out the back in less than a second and the motor shuts off. The card can be as long as you wish to make it. I have designed cards that are 11 inches in length, printed on both sides and requiring two passes through the reader. It all depends on the application.

The units are available with interfacing for the TRS-80, Apple II and the PET. Suitable hookup instructions and software routines are furnished. My work specifically had to do with using the reader as a peripheral for the Commodore PET. One application involved a test scoring program which I will now explain in some detail. A complete listing of the program is shown in Listing 1.

This program was created by Dr. Sam Spero of Cleveland and has been translated for use with the Apple II as well as the PET. It opens with a message explaining that each student's marked card will be scored and that other factual class comparisons will be derived and displayed. The graphic symbols (lines 14, 45, etc.) are PET's unique control symbols for clearing the screen and positioning the cursor. The user is then asked to set the parameters of "number of pupils tested" and "number of questions on the test." An option is then provided whereby the instructor can input responses relevant to missed questions. This provides some guidance to pupils who use this program and are in need of corrective study information.

The program then asks for and accepts the key card. This is the prepared master marked with the right answers. After that, the pupils' cards are inserted, read, scored, and displayed. When all of the students' cards have been thus entered, a display of the summary of all test scores is shown. This provides an overview of class performance. The mean test score is calculated and shown along with the standard deviation figure for the class as a whole. A bar graph (histogram) is also presented which portrays very graphically how the class scoring was distributed.

At this point, if the option has been taken, individual grade reports are displayed along with the previously entered commentaries relevant to each missed question. All in all, the program cleverly provides a fast and efficient means for test scoring and student feedback. The concept is readily adaptable to a broad range of educational and business requirements.

An elaboration on the subroutine which I created for card reading and scoring by the PET will now be discussed. Note that the card illustrated in figure 1 has the answers arranged two to a row. I will later cover the method that can be used to differentiate the answer in the left-hand half of the card from the right-hand side. It is essential to provide various statements and subroutines within the BASIC program which will provide the overall result of reading each row of each card passed through the Mark Reader. It is also desirable to check for such errors as misread timing marks. The values derived for each row read must then generally be decoded into relevant information for the program's usage.

MACHINE LANGUAGE ROUTINE

Due to the speed with which a card passes through the reader, it is not feasible to have a BASIC routine accomplish the reading and error-checking. It is best to employ a machine language routine which executes rapidly when called by PET's SYS command. Such a routine must be POKE'd away into memory prior to calling it for the first time. The subroutine starting at line 10000 is complete and has the POKE routine (line 10005) as well as the converted 6502 operation codes.

When the program is run, 166 bytes of memory starting at location CA (whatever you wish that to be) are altered to contain this card reading routine. One of the best locations for this storage is in the normally unused second cassette buffer starting at address 826. Address 634 (tape buffer #1) can also be used providing you neither read nor write to the cassette during the program's execution. In general terms, this routine reads as many rows as have been previously identified in variable CC%(0). It checks to make certain that the count agrees with the number of timing marks seen by the Mark Reader. If not, then the value of ST (status) is altered and a means is thereby provided for the program to recognize and act on this condition.

The routine further accomplishes the task of passing the cumulative decimal value of the binary representation of each row to an array labelled CC%(X). When this routine has been executed and the return to BASIC is made, the values pencilled on the card may then be found in this array.

"OLD" PET MODIFICATION

Certain addresses in the previously noted routines are unique to the new PET (16K and 32K large keyboard). They must be changed if this routine is to be used in an old PET. The 33 changes to be made are accomplished with the subroutine starting at line 40000.

INITIALIZATION

To make use of the machine language routine it is first necessary to set some values in the program. In the statement at line 10 we have:

1. cleared the PET of all variable values.
2. established the number of rows (or lines) on the scoring card as 55 (or whatever is relevant for your purposes). This is the same as the number of in-line timing marks that are pre-printed in conductive ink on the card.
3. dimensioned an array of that size.
4. set the lowest element in that array to that same value.

IMPLEMENTING LANGUAGE ROUTINE

Line 90 sets the starting address for the machine language routine storage location. In this case, it is the first cassette buffer. Then the subroutine previously described is called and executed. Line 95 calls for the "old" PET modification routine but only if a PEEK at location 50000 reveals that the program is running in an old PET. This statement takes advantage of an idiosyncrasy of the earlier PETs that prevent you from examining the BASIC ROMs.

CARD INPUT MESSAGE

The three statements starting at line 215 inform the user that they are to place a card through the Mark Reader. Line 225 tests for an error and loops back to the beginning after an appropriate message.

CARD INPUT SUBROUTINE

The simple subroutine starting at line 30000 resets the special array (CC%), waits until you have inserted a card, executes the machine language routine, and signals an "OK" if the card was read correctly.

DECODING

For the split row scoring cards, the routine starting at line 6200 translates a pencil mark into the appropriate answer. Lines 6240 to 6300 change a pencil mark on the left hand half of the card into either an A, B, C, D, E or "-" (indicating no answer). Lines 6340 to 6400 do the same for the right half. In line 6230, J represents the actual number of answers to be calculated which may be less than or equal to NL. The X loop starts at 6 rather than 1 because the first 5 rows (in this example) are reserved for the student's ID number.

These routines, for the most part, can be transplanted into other application programs. With a little imagination, the combination of the Commodore PET and the new Chatsworth Data MR500 Mark Reader should prove useful in many ways.□

Program follows

PROGRAM LISTING

```
1 REM        TEST SCORING PROGRAM
2 REM     RE-WRITTEN FOR THE **PET**
3 REM    BY COMPU-QUOTE (213)348-3662
4 REM         FOR CHATSWORTH DATA
5 REM        LAST REVISION 7/30/79
6 REM           VERSION 2.1
10 CLR:NL=55:DIMCC%(NL):CC%(0)=NL
14 PRINT"□"
15 PRINT"THIS ▒TEST SCORING▬ PROGRAM WILL:"
16 PRINT
20 PRINT" 1. SCORE EACH STUDENT'S TEST."
22 PRINT" 2. PRINT STUDENT'S NAME, # OF CORRECT"
23 PRINT"    AND WRONG ANSWERS, TEST PERCENT,"
25 PRINT"    AND A LIST OF PROBLEMS MISSED."
27 PRINT" 3. PRINT A FREQUENCY TABLE OF TEST"
28 PRINT"    SCORES."
29 PRINT" 4. COMPUTE THE MEAN TEST SCORE AND THE"
30 PRINT"    STANDARD DEVIATION."
31 PRINT" 5. PRINT AN ITEM ANALYSIS OF EACH TEST"
32 PRINT"    QUESTION."
33 PRINT" 6. PRINT A BAR GRAPH OF TEST SCORE"
34 PRINT"    DISTRIBUTION."
35 PRINT" 7. PRINT EACH STUDENT A NOTE GIVING:"
37 PRINT"    A) STUDENT'S SCORE"
39 PRINT"    B) # OF QUESTIONS MISSED"
41 PRINT"    C) A BRIEF REMARK ABOUT EACH WRONG"
42 PRINT"       PROBLEM"
45 GOSUB500:PRINT"▩▩▩▩"
60 PRINT"HOW MANY PUPILS WERE TESTED? ";:GOSUB20000:P=VAL(VV$):PRINT"□"
61 PRINT"▩▩▩▩"
62 PRINT"HOW MANY QUESTIONS ON THE TEST? ";:GOSUB20000:J=VAL(VV$):PRINT"□"
63 IFJ>100THENPRINT:PRINT"▒100 QUESTIONS IS THE MAXIMUM":PRINT:GOTO62
68 DIMSN$(P),B$(J)
69 DIMWP(P),TS(J+2),W(J)
70 DIMW1%(P,J)
90 CA=634:GOSUB10000
95 IFPEEK(50000)=0THENGOSUB40000
104 PRINT"▩▩▩▩"
106 PRINT"DO YOU WISH TO INPUT RESPONSES TO MISSED";
107 PRINT"QUESTIONS ?"
108 PRINT:PRINT:PRINT"▒PRESS Y OR N ";
109 GETQ$:IFQ$=""THEN109
110 IFQ$="N"THEN145
111 PRINT"□":PRINT:PRINT
112 PRINT"   TYPE IN THE RESPONSE YOU WISH THE"
114 PRINT"   COMPUTER TO MAKE TO WRONG ANSWERS."
115 PRINT: PRINT"TYPE A ▒(RETURN)▬ WHEN YOU HAVE FINISHED."
116 PRINT:PRINTTAB(9)"DO NOT USE ANY COMMAS."
117 PRINT:PRINTTAB(9)"TYPE 'N' IF NO COMMENT."
119 PRINT:PRINT"  EXAMPLE:":PRINT
120 PRINT"YOU NEED TO STUDY THE MATERIAL ON PAGE"
121 PRINT"57. LOOK AT PROBLEM #5 CAREFULLY."
124 GOSUB500
125 FORS=1TOJ
128 PRINT"□":PRINT:PRINT:PRINT:PRINT
129 PRINT"RESPONSE TO QUESTION #";S
132 PRINT
133 PRINT"?";:GOSUB20000:B$(S)=VV$
135 IFB$(S)=" N"THENB$(S)=" NO COMMENT."
140 NEXTS
145 PRINT"□"
150 PRINT"▒INPUT KEY CARD",
155 GOSUB30000:PRINT:GOSUB6200
160 IFR$<>" "THEN145
170 AN$=L$
200 FORXS=1TOP
210 PRINT"□";
215 PRINT"▒INPUT STUDENT CARD",
220 GOSUB30000
225 IFST>0THENPRINT:PRINT"▒PLEASE RE-ENTER THE SAME CARD":GOTO215
228 PRINT:PRINT"STUDENT ID:";
230 GOSUB6000:SN$(XS)=L$
232 GOSUB6200
234 IFR$<>" "THEN210
259 PRINT"□"
260 FORS=1TOJ
270 IFMID$(L$,S,1)=MID$(AN$,S,1)THEN290
280 W(S)=W(S)+1:WP(XS)=WP(XS)+1:W1%(XS,WP(XS))=S
290 NEXTS
300 C=J-WP(XS):TS(C)=TS(C)+1
320 NEXTXS
340 GOSUB990:GOTO1010
400 PRINT"▩":FORX=1TO23:PRINT:NEXTX:PRINT"▒DEPRESS ANY KEY FOR NEXT PAGE"
405 GETR$:IFR$=""THEN405
410 PRINT"□":RETURN
500 PRINT"▩":FORXT=1TO21:PRINT:NEXT
510 PRINT"▒PRESS (RETURN)"
520 GETA$:IFA$=""THEN520
530 PRINT"□":RETURN
990 PRINT"▒STUDENT"TAB(23)"▒ITEMS"
1000 PRINT"▒# RIGHT";
1001 PRINTTAB(8)"▒WRONG▬  ▒AVG."TAB(23)"▒MISSED"
1005 RETURN
1010 FORS=1TOP
1020 C=J-WP(S):AV=INT(C*100/J+.5):T=0
1030 PRINTSN$(S)
1040 PRINT"   "CTAB(8)WP(S)TAB(14)AV"%";
1045 PRINTTAB(19);
1050 FORN=1TOWP(S)
1055 IFW1%(S,N)=0THEN1090
1060 PRINTW1%(S,N);:T=T+1
1070 IFT<5THEN1090
1071 T=0
1080 PRINT:PRINTTAB(19);
1090 NEXTN
1092 GOSUB400:GOSUB990
1110 NEXTS
1112 PRINT"□"
1115 GOSUB500:GOSUB1120:GOTO1250
1120 PRINT:PRINT"ITEM ANALYSIS":PRINT:PRINT
```

Now! North Star Application Software!

North Star now offers application software for use on the HORIZON! Now you have one reliable source for both hardware and software needs! The first packages available are:

NorthWord—
NorthWord is a simple-to-operate word processing system designed for use with the popular North Star HORIZON. NorthWord enables you to increase office efficiency and cut document typing time and cost. NorthWord incorporates the most sought-after word processing features: easy editing, on-screen text formatting, simultaneous document printing, and much more. NorthWord can be integrated with other North Star software packages to produce customized letters, labels and reports quickly and efficiently.

MailManager—
MailManager enables you to compile and maintain complete organized mailing lists. Lists are easily accessible and can be compiled with a great deal of flexibility. Entries, corrections and deletions are easily made. The North Star MailManager can print your list on individual envelopes, on mailing labels, or in compact summary form.

InfoManager—
InfoManager is a powerful list-oriented, data management system. It will accept up to 50 categories of information for each record and has the ability to select and sort before printing. The North Star InfoManager has power and flexibility for many applications: product inquiry, inventory, customer/client records, calendar reminders, and as an easy way to fill in often-used forms.

GeneralLedger—
General Ledger and Financial Reporting, two programs in one, maintains general ledger accounts based on such input as checks, bank deposits and journal entries, and uses the information in the general ledger to produce customized financial statements and financial reports.

NorthWord is the central building block for all the North Star application software to follow. Packages now being tested include other accounting and professional application packages. For more information or a demonstration, contact your local North Star dealer.

CIRCLE INQUIRY NO. 50

NorthStar

North Star Computers, Inc.
1440 Fourth Street
Berkeley, CA 94710
(415) 527-6950
TWX/Telex 910-366-7001

```
1240 PRINT"▒ITEM #"TAB(7)"▒# CORRECT"TAB(18)"▒# WRONG"TAB(27)"▒% CORRECT"
1245 PRINT:Q=0:RETURN
1250 FORS=1TOJ
1260 C=P-W(S):AV=INT(C*100/P+.5)
1270 PRINT"   "STAB(12)CTAB(20)W(S)TAB(27)AV"%"
1275 Q=Q+1:IFQ<>15THEN1280
1276 GOSUB400:GOSUB1120
1280 NEXTS
1290 PRINT:GOSUB500
1300 PRINTTAB(9)"▒SUMMARY OF TEST SCORES":PRINT
1310 PRINT"▒TEST SCORE","▒# OF TESTS"
1315 PRINT:Q=0
1320 FORS=JTO0STEP-1
1330 IFTS(S)=0THEN1350
1340 PRINTTAB(4)STAB(23)TS(S)
1350 NEXTS
1355 GOSUB500
1356 SU=0:M=0
1360 FORS=1TOJ:M=M+TS(S)*S:NEXTS
1370 M=M/P
1380 FORS=1TOP:D=(J-WP(S))-M:SQ=D*D:SU=SU+SQ:NEXTS
1390 SD=SQR(SU/P)
1400 PRINT"▒":PRINT:PRINT:PRINT:PRINT:PRINT:PRINT
1420 PRINT"THE MEAN TEST SCORE IS"M:PRINT
1430 PRINT"THE STANDARD DEVIATION IS"SD
1440 PRINT:GOSUB500
1500 PRINT"▒":PRINT:PRINT:PRINT:PRINT
1550 PRINT"▒BAR GRAPH OF TEST SCORES":PRINT:PRINT
1552 M=0:XB=1
1555 IFJ>13THENXB=J/13
1556 A=INT(XB):IFA<XBTHENXB=INT(XB)+1
1560 FORS=0TOJ
1570 IFTS(S)>MTHENM=TS(S)
1580 NEXTS
1590 PRINTSTR$(M);
1600 FORS=JTO0STEP-1
1610 IFTS(S)<MTHEN1630
1620 A=3*(J-S)
1624 A=INT((A/XB)+2)
1625 IFS-XB<10THENA=A-1
1628 PRINTTAB(A)"*";
1630 NEXT
1650 PRINT:M=M-1:IFM<1THEN1680
1660 GOTO1590
1680 PRINT
1685 FORS=JTO0STEP-XB
1689 IFS<10THEN1694
1690 PRINTSTR$(S);
1692 GOTO1700
1694 PRINT" "STR$(S);
1700 NEXT
1710 PRINT:GOSUB500
1800 IFQ$="N"THEN1864
1801 V=1
1802 FORS=1TOP
1803 ONVGOTO1804,1807
```

```
1804 PRINT"▒":FORX9=1TO11:PRINT:NEXTX9:PRINTTAB(14)"▒GRADE REPORTS"
1805 PRINTTAB(14)"**************":GOSUB500:PRINT"▒"
1806 PRINT:PRINT:PRINT:PRINT:GOTO1811
1807 PRINT"▒":FORX9=1TO21:PRINT:NEXTX9
1808 PRINT"▒PRESS (RETURN) FOR NEXT STUDENT REPORT.";
1809 GETA$:IFA$=""THEN1809
1810 PRINT"▯":PRINT:PRINT:PRINT:PRINT
1811 PRINTTAB(5)"STUDENT # "SN$(P)", PRESS (RETURN)"
1812 PRINTTAB(5)"TO DISPLAY YOUR TEST RESULTS."
1813 GETA$:IFA$=""THEN1813
1814 PRINT"▯":V=2
1815 AV=INT((J-WP(S))*100/J+.5)
1820 IFWP(S)=0THEN1870
1825 PRINT"STUDENT # "SN$(S)" YOUR TEST SCORE IS"AV"%"
1830 PRINT
1835 PRINT"AND YOU MISSED THE FOLLOWING QUESTION/S:";
1836 PRINT:PRINT
1840 FORX=1TOWP(S):PRINTW1%(S,X);" ";:NEXT
1842 PRINT:T=0:O=4
1844 FORX=1TOWP(S)
1846 PRINT:PRINT"QUESTION #"W1%(S,X)
1850 N=W1%(S,X):PRINTB$(N)
1858 T=T+1:IFT=OTHEN1865
1860 NEXTX
1861 PRINT"▒":FORTX=1TO21:PRINT:NEXTTX
1862 NEXTS
1864 END
1865 T=0:IFX=WP(S)THEN1860
1866 PRINT"▒":FORTX=1TO21:PRINT:NEXTTX
1867 PRINT"I'M AFRAID THERE'S MORE ":PRINT"PRESS (RETURN)"
1868 GETA$:IFA$=""THEN1868
1869 PRINT"▯":O=5:GOTO1860
1870 PRINT:PRINT:PRINTTAB(13)"▒CONGRATULATIONS ":PRINT
1875 PRINT"YOU MADE A 100% ON THE TEST.":GOTO1861
6000 REM-----------------GET STUDENT ID
6005 L$=""
6010 FORD=1TO5
6030 Z2=CC%(D)
6040 Z1=Z2/2:Z2=0
6050 IFZ1>2THENZ1=Z1/2:Z2=Z2+1:GOTO6050
6060 L$=L$+RIGHT$(STR$(Z2),1)
6070 NEXTD
6080 PRINTL$:RETURN
6200 REM------------------CREATE AN ANSWER STRING
6210 L$=""
6230 FORX=6TOJ/2+5.5
6235 C$="*"
6240 N=CC%(X)AND62
6250 IFN=0THENC$="-"
6260 IFN=2THENC$="A"
6270 IFN=4THENC$="B"
6280 IFN=8THENC$="C"
6290 IFN=16THENC$="D"
6300 IFN=32THENC$="E"
6310 L$=L$+C$
6320 C$="*"
```

```
6340 N=CC%(X)AND3968
6350 IFN=0THENC$="-"
6360 IFN=128THENC$="A"
6370 IFN=256THENC$="B"
6380 IFN=512THENC$="C"
6390 IFN=1024THENC$="D"
6400 IFN=2048THENC$="E"
6410 L$=L$+C$
6415 NEXTX
6420 FORD=0TOJSTEP5
6430 FORY=1TO5
6440 D$=""
6450 IFD+Y=J+1THEN6530
6460 IFD+Y<10THEND$="  "
6470 IFD+Y>9ANDD+Y<100THEND$=" "
6480 PRINTD$;
6490 PRINTD+Y;MID$(L$,D+Y,1);" ";
6500 NEXTY:PRINT:NEXTD
6530 PRINT"☒":FORX=1TO22:PRINT:NEXTX
6532 PRINT"☒DEPRESS SPACE BAR TO CONTINUE. DEPRESS"
6534 PRINT"☒ ANY OTHER KEY TO RE-ENTER LAST CARD. ";
6536 GETR$:IFR$=""THEN6536
6540 RETURN
10000 REM------------------READ A CARD
10005 FORX=0TO165:READY:POKECA+X,Y:NEXT:RETURN
10010 DATA120,216,160,0,169,195,209,44
10015 DATA208,5,200,209,44,240,7,169,64,141,150,0
10020 DATA88,96,160,6,177,44,200,200,209
10030 DATA44,16,7,169,32,141,150,0,88,96,133
10040 DATA165,198,165,177,44,133,164,169,204,141
10050 DATA76,232,200,162,0,134,160,134,161,169
10060 DATA16,44,18,232,240,84,44,79,232,16
10070 DATA79,112,242,173,79,232,162,236,142,76
10080 DATA232,73,63,5,161,133,161,173,79,232
10090 DATA162,204,142,76,232,73,63,5,160,133
10100 DATA160,44,79,232,16,44,80,221,165,160
10110 DATA10,10,70,161,106,70,161,106,170,165
10120 DATA161,41,15,145,44,200,138,145,44,200
10130 DATA198,164,48,16,198,165,16,7,169
10140 DATA16,141,150,0,88,96,44,79,232,48,159
10150 DATA169,0,141,150,0,165,164,240,5,169,1,141,150,0,88,96
20000 REM------------------INPUT ROUTINE
20010 V=1:V$="":VV$=" "
20020 GETV$:IFV$=""THEN20020
20030 PRINTV$;:IFV$=CHR$(20)THENV=V-1:VV$=LEFT$(VV$,V):V$="":GOTO20020
20040 V=V+1:VV$=VV$+V$
20050 IFV$<>CHR$(13)THEN20020
20060 IFVV$=" "+CHR$(13)THEN20010
20070 VV$=LEFT$(VV$,V-1):RETURN
30000 REM------------------CARD INPUT ROUTINE
30010 FORZ=1TONL:CC%(Z)=0:NEXTZ
30020 WAIT59471,128:SYS CARD
30030 IFST=0THENPRINT"☒O.K."
30040 RETURN
40000 REM------------------CHANGE TO OLD PET
40010 FORX=1TO33:READY,Z:POKECA+Y,Z:NEXT
40015 RETURN
40020 DATA7,126,12,126,18,12,19,2,25,126
40030 DATA29,126,35,12,36,2,40,53,42,53
40040 DATA44,126,46,52,56,48,58,49,84,49
40050 DATA86,49,98,48,100,48,109,48,113,49
40060 DATA116,49,120,49,124,126,128,126
40070 DATA131,52,135,53,141,12,142,2
40080 DATA153,12,154,2,156,52,162,12,163,2
63000 REM------------------PROGRAM END
READY.
```

System of the Month

The Escon Selectric Interface

By Hampton G. Miller and Andrew Klossner

An IBM Selectric typewriter can serve as an output device for almost any computer when interfaced via Escon's Selectric Converter Model E-A. This product includes a factory assembled universal adapter, together with a modification kit which enables the mechanical operation of the typewriter to be controlled by a program. The kit is easily installed in a few hours; no drilling or other permanent modification is made to the Selectric. The adapter can be any one of a number of standard output devices, eliminating the need for special hardware or software.

The Selectric has established a solid reputation in the field of office typewriters. With the classic golf ball typehead replacing the usual typewriter keybars, a wide variety of type fonts and special character sets are available. At slightly over ten characters per second, a Selectric cannot match speed with the faster daisywheel or dot matrix printers, but it is an excellent output device.

The universal interface consists of a small box of electronics, including a 6502 microprocessor, which accepts ASCII characters from the computer and translates them into mechanical movement of the elements within the Selectric. Options for connecting the universal interface to a computer include TTL, 20 ma, and RS232 serial interface, with three handshake protocols and sixteen baud rates; TTL parallel with three handshake methods and selectable polarity; and two IEEE-488-1975 modes. An interface unit which plugs directly into an S-100 bus is also available. The interface is a separate unit from either the Selectric or computer; it is connected by cable to both, and draws power from the typewriter.

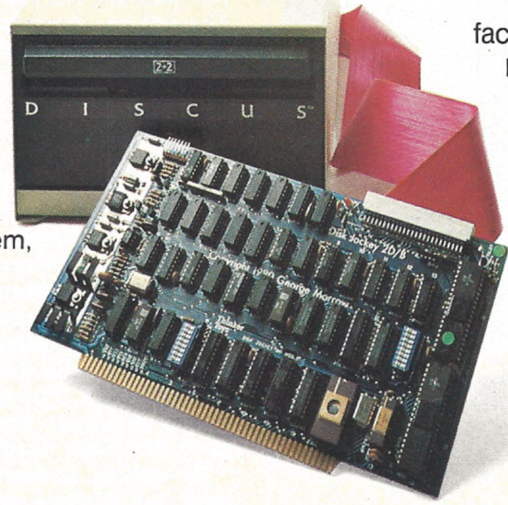
The first modification step is to cut and strip the electromagnet wires. Then three rods controlling shift interlock, switch interlock, and return interlock are removed from the typewriter to make room for the electromagnet assemblies. Next the "select" and "function" electromagnet assemblies are installed, and the three interlock rods are replaced. The "case shift" electromagnet assembly is installed; then all electromagnet wires are spliced into the cable to the interface unit, and the cable is tied to the frame. Now an ohmmeter test is conducted. When it passes, the unit is ready for power and can be tested by the interface unit.

To connect the interface to the computer, the interface and handshake methods must first be chosen. Handshaking occurs when the interface buffer becomes full or when it later empties. When the buffer becomes full, the interface signals the computer that it must stop sending characters; as the buffer empties, it signals the computer that more characters may be sent. These signals may be in the form of logic lines which change

There are two sides to our story.

Side One

The DISCUS™ 2+2 Quad-Density Hardware



Now you can use your S-100 system to tackle big jobs. Because the DISCUS™ 2+2 Quad-Density Disk System puts 1.2 megabytes of fast-access memory on your side for just $1545.00 complete.

With the DISCUS™ 2+2 System, complete means complete.

You get a full-size (IBM-compatible 8") double-sided/double-density disk drive, factory mounted in a cabinet with power supply, fully-buffered S-100 single-board controller, and interconnecting cables. All fully assembled, system-tested and fully warranteed.

You get the speed and efficiency of 1.2 megabyte-per-diskette memory... and you get it for 0.13¢ per byte.

Side Two

The DISCUS™ 2+2 Quad-Density Software

1.2 megabyte quad-density hardware is only one side of the story. The DISCUS™ 2+2 System price includes all the fully-interfaced, high-performance software you need to take full advantage of your quad capacity.

The system includes our exclusive BASIC-V™ virtual disk BASIC, which allows you to address your quad-density diskettes as easily as main memory. The operating system you get is the widely accepted CP/M* 2.1. And you get our powerful DISK-ATE text editor/assembler; The most advanced software development tool available.

Micro-Soft BASIC 5.1 and Micro-Soft FORTRAN are available as options. Both run under CP/M* 2.1.

Check out the full system price of DISCUS™ 2+2 Quad against any other floppy disk system at your local computer store. At $1545.00, we think you'll take sides with DISCUS™ 2+2.

If your dealer doesn't carry THINKER TOYS products, write MORROW DESIGNS Inc., 5221 Central, Richmond, CA 94804. Or call (415)524-2101 9-5 weekdays (Pacific Time).

MORROW DESIGNS™/ Thinker Toys™

*CP/M is a trademark of Digital Research.

state when the signal is being sent (hardware handshaking), or characters which are sent to the computer (software handshaking). Most peripheral interfaces use one of these methods, so there should be little work involved in configuring the computer hardware or software for Selectric operation.

PERSONAL EXPERIENCE

One of the authors used the Escon system to interface a vintage Selectric to a Heath H11 (LSI-11 based) computer. Modifying the typewriter and connecting the interface took less than eight hours. A TTL level parallel interface (which previously ran a paper tape reader/punch) was used, and the standard paper tape punch software properly drove the terminal.

SELECTRIC OPERATION

When a Selectric is manually operated, a key is pressed and a character is typed. As the key is depressed, code bails are coaxed into position and engage latches to select one of four possible tilts and eleven possible angles of the typeball. The shift key reverses the typeball, giving eleven new angles, for a total of 88 possible characters. After the latches are set, the power cam is actuated, causing the typeball to tilt, rotate, and strike the paper through the ribbon. The typeball carrier then advances one column to the right and is ready for the next character. The non-printing controls include the shift, carrier return, index (line-feed), space, and tab keys.

Under control of the Escon interface, the code bails are directly manipulated and the print cycle is initiated using electromagnets. The return and space functions are also performed in this manner. Another electromagnet assembly controls the shift operation. The microprocessor handles the exact timing requirements for carrier return, shift, and between-character delays, and overlaps shifting with spacing when possible. It contains a 511 character buffer to allow for surges of output; handshaking suspends computer output when the buffer needs time to empty.

As well as standard ASCII (including upper and lower case), the interface responds to special codes which can suspend output to allow manual typing or typeball changing, sound an optional alarm, stop typing and clear the internal buffer, and continually print the contents of the buffer. The interface methods and self test off-line pattern generation are controlled by DIP switches in the interface unit.

INSTALLATION

To assist in the modification of the Selectric, the unit includes a 37-page assembly manual, a 23-page book of illustrations, and a 28-page reference manual. Escon will perform the modification for labor and shipping costs, promising to complete within two weeks.

CAVEATS

A few cautions regarding the system's limitations are in order. Unlike the IBM 2741 computer terminal based on the Selectric design, the Escon/Selectric unit does output only. Pressing keys will cause typing to occur, but will not transmit any information back to the computer. Thus, the system is a printer, not an interactive terminal.

The system does not perform the tab, backspace, or return-without-index operations. This reduces its usefulness for word processing where backspace is required for underlining.

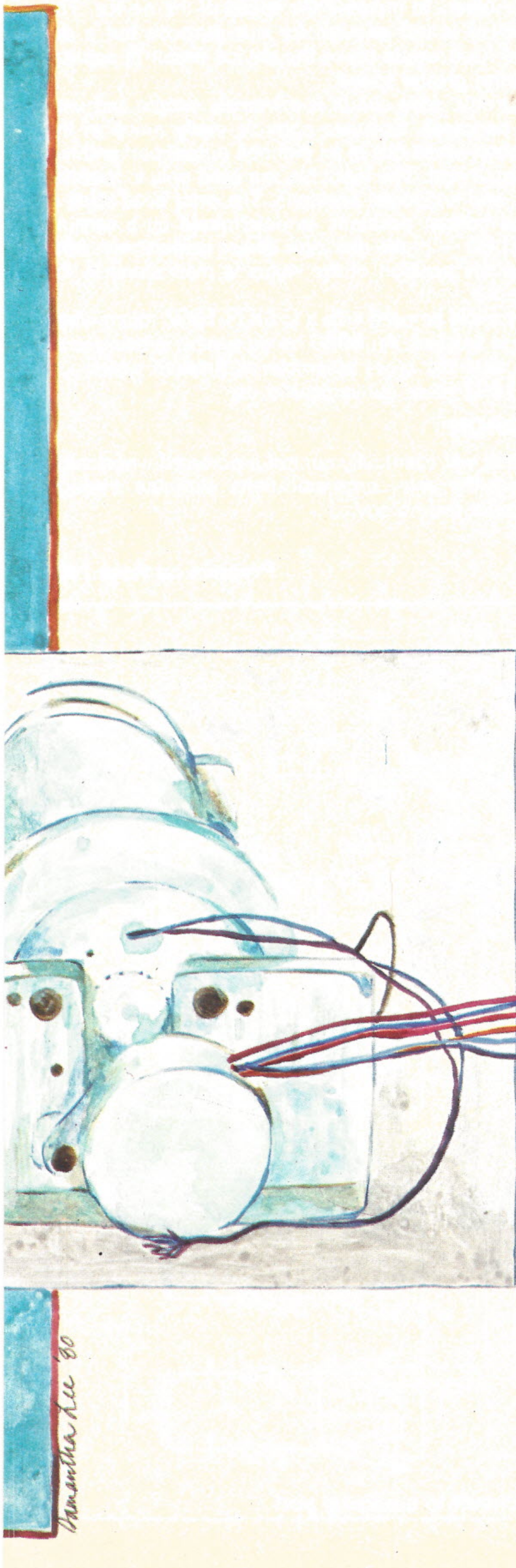
Experience suggests that a Selectric without an impression control would not be as easily converted as the later models.

Finally, if a Selectric is turned off while it is typing, it may become damaged.

CONCLUSIONS

The Escon system is easy to install, interfaces to almost any computer, and provides reliable, professional looking hard copy at a low cost. It is an excellent investment for a personal computer hobbyist or a business person with a small system.☐


How's your love life?
INTERLUDE #99 THE ULTIMATE EXPERIENCE...
A little dull around the edges? Routine? Predictable? Boring? Maybe all it needs is a little Interlude. Interlude is the most stimulating computer game ever conceived. It combines a computer interview, an innovative programming concept, and a one-of-a-kind manual to turn your love life into exciting, adventurous, delicious fun!
CIRCLE INQUIRY NO. 39
Interlude is: romantic . . . playful . . . outrageous . . . a fantasy. Interlude is: ■ Wet fun on a hot summer night. (Interlude #21) ■ A surprise on the way home from dinner. (Interlude #42) ■ A bubble bath that ends with a bang. (Interlude #78) ■ An evening to rest while she does all the "work." (Interlude #25) ■ The most romantic of evenings. (Interlude #84) ■ A new twist to an old subject. (Interlude #69) ■ Just watching her . . . (Interlude #57) ■ An erotic fantasy! (Interlude #33)
With over 100 Interludes, you can satisfy all levels of interest and desire. Each Interlude is fully described in the manual, and the more elaborate ones are detailed with regard to settings, props, and mood-enhancing techniques. But we've saved a few super Interludes for that very special time when your interview indicates you're ready! At that time, you will be introduced to one of several Interludes held secret within the computer. (When you learn secret Interlude #99, your love life may never be the same again!) Interlude can give you experiences you'll never forget. Are you ready for it?
Interlude™
The Ultimate Experience.
Interlude, 10428 Westpark, Houston, Texas 77042 I'm really ready! Rush me ______ copies of Interlude today.
☐ For the Apple II (16K) # ☐ For the TRS-80 (Level II-16K) ## ☐ $14.95 for cassette ☐ $17.95 for diskette.
Add $1.50 for shipping. Texas residents add 6% sales tax. My check (payable to Interlude) is enclosed. ☐
*Charge my ☐ MASTERCHARGE ☐ VISA account.
Account No. ______ Expiration date ______
All charge customers must sign. Signature ______
Name ______ Age ______
Address ______
City ______ State ______ Zip ______
*CHARGE CUSTOMERS: Order by phone toll-free! 1-800-327-9009 Ext. 306
Apple II is a registered trademark of Apple Computers, Inc. ## TRS-80 is a registered trademark of Radio Shack, a Tandy Co.



The Dust Writer

By Michael J. Hodgetts
University of Tennessee

At the University of Tennessee Rehabilitation Engineering Center in Memphis, we work with severely handicapped children to find ways to get around the effects of their handicaps. Alaine Marty is a little girl who has cerebral palsy which prevents her from using her legs, arms and vocal organs.

She communicated with her teachers by eye movements, looking left for yes and right for no. But a faster way to communicate that would not require an extra person's cooperation was needed. The Electronics Department was asked to adapt a new electronic device, called a TIC, which was developed at Tufts-New England University Medical Center.

With this device, a switch is closed once to select one of several rows of characters. The scanner then stops in that row, and the user hits the switch a second time to select a character from the row. The character is then displayed on a small CRT.

We were asked to replace the switch with a photo-cell that could be operated by a head mounted light-stick (a special type of flashlight). Since it is hard to look at a letter and then move to point the light at it, the final solution was a head-mounted mercury switch. But in the process of her trying the aiming method it became obvious that she could aim the light very accurately.

An idea formed. Why not wear a head mounted light pen and use a screen two feet away as a keyboard? For low power and portability the screen was constructed from sixty-four LEDs and the light pen was designed to respond to the fast rise time of the light from the pulsed LEDs. This new device permits her to communicate much faster than ever before and she may compose messages or school work on an output device without anyone else helping. She calls the unit Aunt Martha.

Aunt Martha uses a CRT terminal for output and is not portable. For the system to be portable, a lightweight device is needed. It must also draw very little power, be readable in

daylight and must be low in cost. For this purpose it should display at least three or four lines of thirty-two or more characters. In the interest of safety, voltages should be kept below thirty volts. The device should have the potential for graphics display as well as upper and lower case characters.

Let's look at the technology available.

1. Cathode-Ray Tube Terminal
   Too much power consumption, weight and size. Also a CRT uses high voltages, and washes out in daylight.
2. Neon
   Again too much power consumption, high voltage and daylight washout.
3. Vacuum Fluorescent
   Power consumption is lower in small displays but daylight washout is still a problem and large displays are not readily available.
4. Light Emitting Diode
   Power consumption is too high and daylight viewing is not good. Also the cost for a large display is excessive.
5. Liquid Crystal Display
   Someday this may be the answer but for now the cost, availability, and driver complexity make it impractical.

To give the system some mobility, we designed a device that is lightweight and draws very little power. It can be mounted on the front of a wheelchair with little trouble.

The Dust Writer draws no power except when actually writing a new character. It is lightweight, small, inexpensive, and may be viewed in bright light.

The principle of operation is the same as that of the Etch-A-Sketch® toy made by Ohio Art. We actually used the powder from an Etch-A-Sketch toy in our device. The configuration is that of a drum plotter with the stylus on the inside of a glass drum. A stepping motor drives the stylus horizontally with a threaded shaft. Another stepper drives a cam for vertical motion and a solenoid lifts the stylus from the glass when necessary. Line feed is accomplished with a small D.C. gearmotor that turns the drum. The powder in the bottom of the drum erases the old printing so that fresh media is always fed up to the drawing field.

Many mechanical arrangements are possible and we plan to try some others to increase the speed of the device. The present system is fast enough for our purpose but a dot matrix print head would make the device useful in applications requiring greater speed.

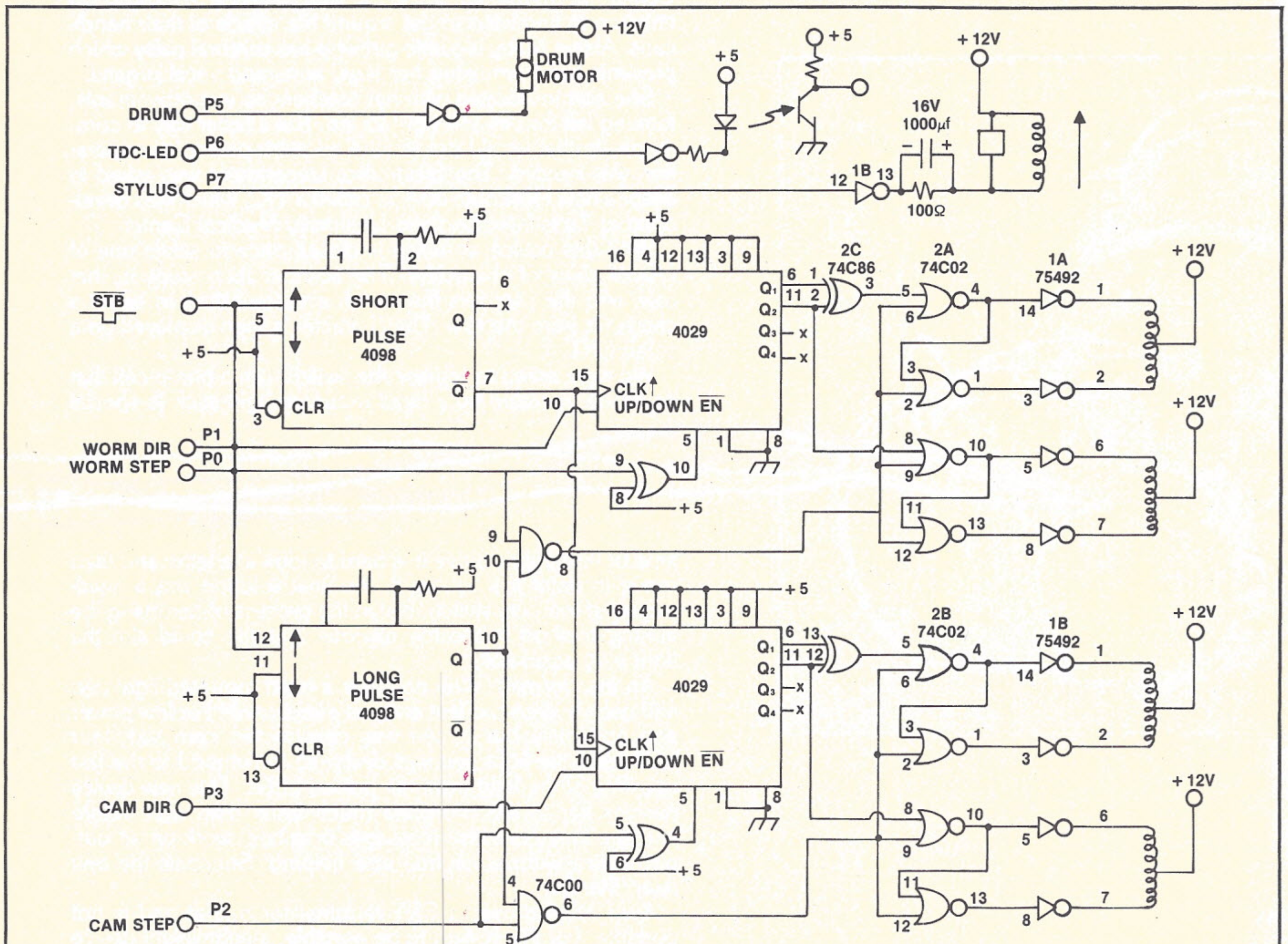
The electronic drive circuit is extremely simple and uses only nine packages. Software controls every move of the device through seven bits of an eight-bit output port and two bits of an input port. The input bits are used only to initialize the vertical and horizontal positions. The print head starts at the left and bottom positions as determined by a sensing switch and LED — photo transistor device respectively. After initialization the position of the stylus is maintained in the microcomputer.

The device will be used with a C-MOS 1802 micro when the system is finished. An 8080 based system is being used for testing until the cross assembler is finished which will make the 1802 more convenient to use.

CONCLUSION

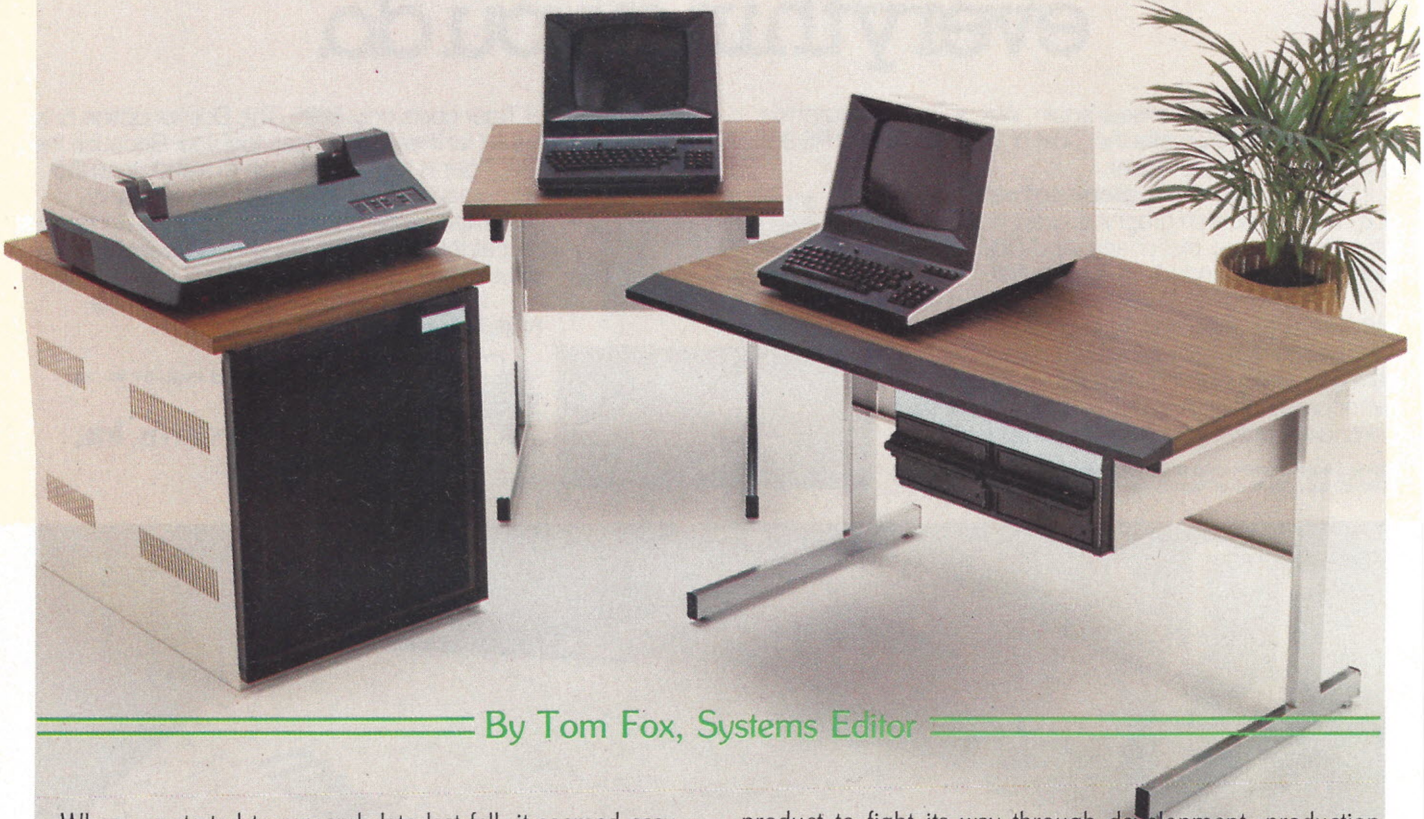
The system will eventually control a powered wheelchair, making mobility and communications available through microcomputer technology.□

Program on Page 140



Electronic interface and stepping motor drive circuit to a parallel port.

Looking at Micro-Based Business Systems



By Tom Fox, Systems Editor

When we started to research late last fall, it seemed easy enough: write an article comparing all of the available microcomputer systems. Even living in the computer industry, we were unprepared for the huge number of machines that exist, and the complexity contained within each one. Understanding how even one computer works, and unearthing its weak and strong points, is a week's work. Doing it for 30 or more systems, while trying to find some common ground for comparison among them, gave us cause for thanks that we only do this once a year. It made us appreciate some of the headaches the ultimate purchaser must endure in selecting a computer system.

PICKY, PICKY

A question you may fairly ask is: How did we arrive at the particular choice of computer systems represented here? In large part, we selected them the same way you might: looking at magazine advertisements, poking around in computer stores, even answering a radio commercial. In researching this subject, we visited over 20 computer stores in half a dozen cities in three states. We wanted to see what systems were actually available to a retail purchaser. We limited our consideration to those systems that survived multiple traumas of conception, development, production and distribution to reach the retail level.

We realize that in demanding such stiff qualifications, we would be limiting ourselves to machines from last year's state-of-the-art; it takes at least that much time for a new product to fight its way through development, production and distribution. Because of this, we have included two or three of the most promising new computer systems that offer something special in the way they work or are being marketed.

Large or small, sophisticated or simple, we tried to pick a representative sample of available equipment. It was clearly impossible to include them all.

We are emphasizing microcomputer systems intended for use in small business situations. Where there was a choice, we picked a system that utilized a hard disk drive as its primary storage medium. Although we've seen a lot of powerful floppy disk-based small business systems, we think that a hard disk drive in the ten-megabyte or larger size is more suitable to the bulk of serious business computing.

CATEGORICALLY SPEAKING

The systems can be categorized in many different ways: number of bits handled at once by the microprocessor (eight or 16), bus compatibility (S-100, SS-50, etc.), capacity or type of disk (floppy or hard), single- or multiple-terminal, BASIC- or Pascal-based, types of application software supplied and, of course, price.

Let's begin with computers utilizing eight-bit microprocessors, followed by the generally higher-performance and more expensive 16-bit units. We can further break down the eight-bitters into the actual design of microprocessor utilized, since there is a certain amount of software compatibility within the groups, and only a limited amount of program interchangeability outside of each family.

By far the most popular microprocessor used in small business computer systems is the 8080 and its derivatives, the 8085 and Z80. The latter two operate at a faster rate than grandfather 8080, and the Z80 sports an enhanced instruction set at the machine-language level. There is a great deal of already-written software that will run on all three. The second group of eight-bit systems we will look at are those based on Motorola's 6800 microprocessor chip. Actually, both of our entries this month use the advanced 6809 version. Finally, we will look at a pair of 6502-based systems. In the industry, this microprocessor design has been somewhat of a sleeper, and would probably be largely unknown if the Apple personal computer had not been such a spectacular success in the marketplace.

Sixteen-bit microcomputers share far less commonality than their eight-bit siblings; systems in this category tend to be more distinguishable from each other. We have two 8086-based designs, one that is a near copy of Digital Equipment Corporation's LSI-11, and three others that are as different from each other as they are from the rest of the entrants.

8080 GROUP OF EIGHT-BIT MICROCOMPUTERS

Altos Sun-Series

Decipher the model number for this product, and you have a thumbnail description of the computer itself. The ACS8000-6/MU4 is an Altos Computer Systems' series 8000 with six serial input/output ports and a multi-user disk operating system configured for four simultaneous users. To keep all of these terminals satisfied, a generous 208 kilobytes (KB) of Random Access Memory (RAM) is fitted inside a tabletop enclosure that also houses the Central Processing Unit (CPU) and dual eight-inch, single-sided, double density floppy disk drives. It takes a separate box to hold the single-platter Winchester-technology hard disk drive and its 14.5 megabytes of data. If more storage is needed, a dual-platter unit can be substituted, and a second single- or dual-platter unit can be added at the same time or later.

The multi-user disk operating system is Altos' own AMEX, which will run CP/M™ compatible programs and languages. Included in the price of AMEX is a hardware floating-point arithmetic board that considerably speeds up mathematical computations. Other than a rather complete selection of programming languages and programmer tools, Altos has chosen to leave the fitting of applications programs to their dealers and end-user customers.

Cromemco System 3

In the November 1979 issue of INTERFACE AGE, we published an in-depth description of the System 3. Since then, Cromemco began shipping these computers with double-density floppy disk drives, without increasing the price for doubling the standard storage capacity to over a megabyte. Cromemco is one of the few survivors of the vicious decimation of S-100 board constructers. They owe their current strength to careful attention to product quality, documentation and innovation in new designs.

The ten-megabyte HDD hard disk add-on is the sealed-media type, so the floppy disk drives have to be retained to extract backup copies of programs and data from the hard disk.

Cromemco offers more variations of BASIC (five at last count) than any one programmer will ever use. All of Cromemco's software, including a unique structured FORTRAN language called RATFOR, is created by an in-house programming staff. If Cromemco is strong in systems software, they are far from being a power in applications programs. Their two releases so far — a small Data Base Management System (DBMS) and two successive versions of a Word Processor — have been greeted with only lukewarm enthusiasm by users. In common with nearly every other computer discussed this month, the retail dealers are only too happy to fill this gap with a wide variety of stock and custom programs that will run on the System 3.

Heath H89

The WH89 is the lightweight of this roundup, both in terms of price and probable utility in a business environment. It is limited to 48 kilobytes of memory and a single mini-floppy diskette drive, although we understand that Heath intends to add a dual 8" floppy disk option later this year. The WH89, with its dual Z80 processors, is certainly fat in the CPU department. The WH89 is actually a WH19 "smart" CRT terminal with a single-board CPU and 5¼" floppy disk drive tucked into the unused corners.

Accessories include three different printers, including an attractively-priced $795 dot matrix device. Software is limited to a disk operating system with Benton Harbor BASIC or the option of the more capable Microsoft BASIC (an additional $100). The operating system is a "must buy" option that will add $100 to the list price.

Heath's new word processing program is so new we haven't seen it running as yet. It lists for $495, but you can take advantage of a $300 discount if it's delivered with the new daisywheel character printer. This latter device is a re-labeled Diablo 1640RO shown in their latest catalog at $2895. Remember when you could buy a Heathkit hi-fi amplifier for $79?

The Heath Co.'s new owner, Zenith Data Systems, has big plans for the WH89-CS. Zenith is well along in its plan to market an identical Z89 throughout the world via established computer stores and franchises. Their goal is to have Z89s displayed at 40% of the computer retail outlets in the United States within six months.

Industrial Micro Systems Series 8000

An article in the December 1979 INTERFACE AGE described the Series 8000 as a solid, middle-of-the-road representative of the 8080 group. Industrial Micro Systems is a hardware manufacturer, and their expertise has made the Series 8000 a robust and reliable computer. The manufacturer depends on outside specialists for software, with a choice of four separate operating systems. CP/M is the most often asked for, and many CP/M-compatible applications have made a cozy home in Series 8000s.

The Series 8000 comes in an Industrial Micro Systems desk-style work station with the best maintenance accessability we've seen. (The manufacturer has made cooling fans standard equipment subsequent to our carping about the lack thereof.) Software is available to take advantage of the extended memory capabilities that have always been a part of the Series 8000 — up to a quarter megabyte in multi-terminal systems. A new 64-kilobyte dynamic memory board is available, breaking a long-time Industrial Micro Systems tradition of building only static RAM cards.

You can now add up to two Control Data 90-megabyte Phoenix cartridge disk drives to a Series 8000. A Phoenix-only system (no floppies) lists for a reasonable $10,000 or so.

Intertec SuperBrain

The SuperBrain is a new breed of desktop computer that appears at first glance to be merely a CRT display terminal. Two double-density 5¼" minifloppy disk drives are barely noticeable next to the display screen, but give up little in capacity: nearly 700 kilobytes if you opt for the QD double-track option. Some 40% of SuperBrain purchasers do just that, feeling the $1,000 premium well spent. If that isn't enough, an 18-megabyte Winchester disk drive is available as an add-on for $4695.

The SuperBrain is a single-board computer, meaning that it does not have the traditional mother board with plug-in slots for the CPU, RAM, etc. It does, however, have space inside for a single S-100-compatible board. This is often occupied by the hard disk controller, but you can add anything you like from the large collection of available products, so long as you curb your desire to a single board at any one time.

The SuperBrain is happiest with CP/M-style programs, and a wide selection of languages and programming aids is available from Intertec which fits that mold. APL should be out in another month. This represents an ambitious undertaking, because it requires an expanded character set for both the keyboard and display screen. Since Intertec controls the production of every part of the computer's terminal, it has the flexibility to add on such an enhancement.

Micro V MICROSTAR

Micro V is an ambitious, two-year-old California company whose corporate roster reads like a veritable Who's Who of movers-and-shakers of this frantic industry. It has taken a careful look at the marketplace (even operating its own retail computer store for a time to get in touch with buyers' requirements) and created the MICROSTAR small business system. It's an 8085-based single-board computer that is normally equipped with a dual floppy disk drive and attractive desk-type enclosure. In common with other computer systems which intend a serious assault on the business market, a hard disk option is available to increase data storage capacity to some 20 megabytes. Micro V allows its dealers to set end-user prices, giving them the freedom to add local software services into the price. But you can expect to pay around $10,000 for an average MICROSTAR system.

The MICROSTAR's strong suit is its software. Its developers have taken the more traditional minicomputer-style approach of serving up a package of powerful programming tools integrated into the system. STARDOS, the multi-user disk operating system, includes an extended business BASIC interpreter with optimized file handling talents. Both sequential and random (direct) files are supported, and tools for Indexed Sequential Access Method (ISAM) are standard on the MICROSTAR. The system features one of the most capable DBMS-type programs (actually more of a stand-alone language) that we have seen on a micro. UPDATE, as it is called, allows data base entry and inquiry with English language commands. It comes standard, too.

Micro V engineers have taken to calling the MICROSTAR the MICROSTAR I in anticipation of the soon-to-be-announced MICROSTAR II, an enhanced machine based on the 16-bit 8086 microprocessor chip.

MicroDaSys millie

The national computer magazines have been running ads inviting the reader to become an instant computer dealer by purchasing just one "millie" (at a discount price), adding any programs the reader can write or buy, and reselling the resulting system at any price he chooses. Some 6,000 persons responded to a month of ads, so the idea must strike a responsive note in hobbyist souls. Actually, this business practice describes what systems houses — and more lately computer stores — have been doing for longer than microprocessors have been around. MicroDaSys is presenting anyone with the moxie to try it, a factory-supported way to crash into the business on a shoestring. It's a unique method for increasing sales.

The millie itself is a repackaged System-Z, a Z80-based dual floppy disk drive S-100 computer that emphasizes its word processing talents to the extent that the only printer in the catalog is a typewriter-quality NEC Spinwriter. The CRT terminal is a new MicroDaSys design optimized to run either the popular Electric Pencil or ascending WordStar word processing programs. MicroDaSys' only contribution to its fat applications software catalog is the Pencil Sharpener and Star Brightener. These are similar packages that enhance the basic word processing programs to allow the merging of mailing lists with text files to create computer generated "personalized" form letters. Think of millie each time you empty your stuffed mailbox in this election year.

North Star Horizon

North Star, four years old now, is an old timer in the world of S-100, Z80-based computer systems. In that time, it has shipped an incredible 10,000 computers and systems. To many, the Horizon is "the one in the wood box." In truth, there is little else to distinguish it from many of the current crop of microcomputer systems. These words are not intended to slight North Star. After all, just where did all these "me too" computers get their best ideas?

The HRZ-2-64K-D was selected by North Star as its most popular configuration. It includes 64 kilobytes of RAM, dual double-density 5¼" floppy disk drives with a total capacity of 360 kilobytes of data storage, a Soroc CRT terminal, and Anadex dot matrix impact printer. For an extra $1920 you can substitute the excellent NEC Spinwriter thimble-type character printer. A new 18-megabyte Winchester hard disk drive has recently been made available. If applications run to large data files with floppy disk storage, opt for the quadruple-capacity minifloppy drives and fit an extra pair for a total capacity of nearly 1.5 megabytes.

North Star DOS enjoys a popularity which challenges that of CP/M as a microcomputer operating system. Almost any computer store with a collection of software features many useful programs that were designed specifically to run on the Horizon. North Star itself does not offer any end-user applications programs yet, but look for developments in this line.

Pertec PCC 2000

Pertec is the giant computer peripherals manufacturer that acquired Altair soon after the latter company unleashed the world's first really popular microcomputer. The PCC 2000 is its direct descendent, and the most successful to date. It features a thoughtfully integrated all-in-one package that shows evidence of careful planning in the keyboard. The microprocessor is the 8085, which runs 50% faster than the 8080s which graced Altair's earlier products.

Pertec controls distribution as tightly as it does manufacturing. Only factory-controlled stores can sell the product, and never with competing brands. It needn't be so concerned. When compared with the current crop of widely-available small business computer systems, the PCC 2000 holds its own.

Radio Shack TRS-80 Model II

Here's the monster in the lineup, if only because of the formidable distribution network of over 7,000 retail outlets. We know a computer accessories manufacturer who received a multi-million dollar order — the biggest in that company's history — to place just one floor sample of his product in each Radio Shack store. This tremendous base of retail outlets was solely responsible for making a mediocre product — the original TRS-80 — the most widely produced computer in history. Well in excess of 100,000 have been shipped.

The follow-up Model II is being marketed far more conservatively in only 150 Radio Shack computer centers and select stores. However, not a single one of the other 6,850 outlets will turn down an order.

The Model II utilizes the same eight-bit Z80 microprocessor chip as the Model I, but it's stoked up a bit more in the younger product — 4 megaHertz as opposed to a lackadaisical 1.78 megaHertz. The Model II comes with a single 8" floppy diskette drive as standard, and we recommend that you fill up the RAM space to a full 64 kilobytes right away. None of the Radio Shack-supplied applications programs will run in the smaller 32-kilobyte version. The disk drive is the

PROGRAM EPROMS WITH YOUR APPLE

NEW FROM MOUNTAIN HARDWARE ROMWRITER™

FASTER THAN A SPEEDING DISK!
Firmware in ROM is as fast as your software in RAM, but in ROM it frees up RAM memory space for companion programs. And, there never needs to be a LOAD from disk! Create firmware for your Apple* by programming EPROMs with RomWriter.

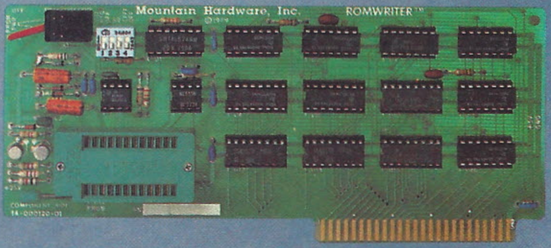
FIRMWARE NOT SOFTWARE.
Many frequently used programs really ought to be installed as firmware. ROM-based firmware permits a "power up and go" configuration. Use RomWriter to create firmware for peripherals such as printers or create program cards. By installing EPROMs that you have programmed on Mountain Hardware's ROMPLUS+™ board, program cards of up to 12K in length may be created.

FEATURES.
Programs 2716 EPROMs (5V). All 2K or part of the EPROM can be programmed. Installs in a peripheral slot. EPROMs mount in a zero insertion force socket. A switch turns power off to the socket so EPROMs can be inserted or removed without powering-down your computer. A Write Protect switch is provided for programmed EPROMs while running. A $CFFF OFF switch to suppress this command during programming or RUNning. EPROMs can be RUN on RomWriter, or ROMPLUS+™ when creating firmware systems.

SOFTWARE
Virtually foolproof programming. Specify a Start and End address in the EPROM and either a Disk File name or a starting address in memory. Desired code will be BURNed followed by a VERIFY. Additionally, existing EPROM code can be merged with desired changes to facilitate EPROM debugging. Easy data entry and high reliability are designed into RomWriter. Programmed EPROMs can be RUN while residing on RomWriter or can be transferred to Mountain Hardware's ROMPLUS+ board, requires Applesoft firmware.

THE RIGHT PRICE.
See for yourself how firmware can enhance the power of your Apple system. Drop by your local computer store for a demonstration. RomWriter and Mountain Hardware's full line of computer products are available at dealers worldwide.

*Apple is a trademark of Apple Computer, Inc.



Available through dealers worldwide

Mountain Hardware, Inc.
LEADERSHIP IN COMPUTER PERIPHERALS
300 Harvey West Blvd., Santa Cruz, CA 95060
(408) 429-8600

- Fast Firmware? Send me all the details on RomWriter.

Name ____________________

Address ____________________

City ____________ State ____________ Zip ______

Phone ____________________

dual-density variety, giving nearly half a megabyte of storage capacity. Radio Shack should be commended for including a little-publicized fact in their catalog: 15% of the first disk's capacity is required to hold system-related programs, and is not available for data applications program storage. On some systems we know, this figure runs closer to 80%.

Additional disk drives are housed in a separate cabinet that can be optionally installed in the pedestal of an integrated systems desk. A good selection of printers is available, including a brand new daisywheel character printer for word processing applications. The word processing program itself will come along in a few months.

The new Level III BASIC is an enhanced version of the Model I's Level II language (just keeping the rows of Roman numerals straight requires a computer). A generous repertory of 114 different BASIC and TRSDOS commands are at your fingertips with Level III, Model II. As for applications programs, only the general ledger, inventory control and mailing list are fully released. Accounts receivable has been recalled by the factory, and payroll was being released at press time.

TEI 3400

TEI is a Houston, Texas maker of transformers and sheet metal products which earned an early reputation for making the best S-100 "box" in the industry. The manufacturer filled those boxes with computer cards and other parts, and it came to pass that a complete machine appeared under the TEI banner: the 3400 Business Computer System.

In talking with the factory, we get the impression that it hasn't firmly decided what direction its product should take. Early advertising featured 50-megabyte Calcomp disk drives and a snazzy two-piece CRT display terminal — both replaced by other products before the first systems rolled off the assembly line. In the software area, the initially-promised proprietary multiprocessing operating system and comprehensive business information management system have been replaced with the considerably toned-down statement that the product is CP/M compatible.

Plans are to offer a proprietary 150 character-per-second printer, as well as several languages (BASIC, FORTRAN, COBOL), in addition to making up the lost time in the business software development project. It could happen. The company still makes the best S-100 boxes around.

Vector Graphic MZ

People always confuse Vector Graphic with North Star, even though the two companies are separated by most of California. Both started the same year with S-100 board products (CPUs, memories, interface boards, etc.), and soon integrated them into their own boxes with a pair of vertically-mounted minifloppy disk drives on the right-hand side. Both companies have been very successful, and recently reported shipping some 10,000 computers out the door in the past four years. The similarities go further, but stop abruptly when one begins to look at Vector Graphic's System B.

This is the model MZ Z80 microcomputer system enhanced with a terminal of its own design and a disk operating system (CP/M) complete with Level V Microsoft BASIC. The CRT display terminal goes by the humble name of "mindless," which says that it does not have the smarts to decipher a character and display it on the screen in the usual fashion. Instead, the terminal depends on a separate S-100 board (supplied with the System B) mounted in the computer chassis. One advantage is that the terminal runs in the memory mapped mode, and can reflect display changes far more rapidly than even the fastest of ordinary "dumb" or "smart" terminals.

Software includes a screen-oriented text editor for programmers; an even more advanced word processor is available for the business user. The well known Peachtree family of business programs (general ledger, accounts payable, accounts receivable, payroll and inventory control) comes standard at the quoted price, but you must pay extra for the printer required to utilize them.

Zilog MCZ-1/70

Here's an interesting story. Zilog — the supplier to the world of the ubiquitous Z80 microprocessor — has decided to come out with its own computer system and compete directly with its own best customers. The idea is not all that new, but it has never been carried off successfully. Years of effort in marketing industrial minicomputer systems has hardly made Texas Instruments a power in that field, and National Semiconductor recently cried "uncle" in its attempt to sell business computers. Both of these companies are giants in the semiconductor industry. Why have they failed to make a mark by putting their chips together into a stand-alone system? It's a fascinating question, but too long to go into here. Today's news is that Zilog is treading the same mine field, and depending upon its new MCZ-1/70 to carry the day.

The MCZ-1/70 is billed as a multiterminal COBOL business computer, although BASIC, FORTRAN, Pascal and PLZ are listed as available languages. The primary thrust is to provide a vehicle for the thousands of COBOL programs that are running on aging number crunchers. The multiterminal capability (available only with COBOL) is just icing on the cake.

A reliable 10-megabyte cartridge disk drive is the central data storage device, but standard-size floppy disk drives are available if you need them. Memory is limited to 64 kilobytes, which must make things a little tight when all five CRT terminals are alive. The terminals are Lear Siegler ADM-31 "smart" terminals with custom firmware programming.

Although Zilog is still sorting out their marketing distribution plans, a visit to their factory convinced us that they are fully committed to producing a great many of these systems.

6800 SERIES OF EIGHT-BIT COMPUTERS

Smoke Signal Broadcasting Chieftain

The 9822 is a model of the Chieftain line that features a 6809 microprocessor board, 48 kilobytes of Random Access Memory and a floppy disk controller in a nine-slot tabletop housing. The boards conform to the SS-50 bus protocol which was initially developed for the Motorola 6800 processor. This particular Chieftain also comes with a dual full-size floppy disk drive having a total capacity of nearly two megabytes. An additional pair of such drives can be fitted, along with the unlikely combination of four 5¼" minifloppy disk drives, yielding up to 7.5 megabytes of storage total. For a hard disk, Smoke Signal offers the clever Honeywell hard disk drive that holds ten megabytes of information in its small 8" removable cartridge. Using the new 32-kilobyte RAM cards, up to 192 kilobytes of memory can be fitted into the computer box.

The latter option will be needed when Smoke Signal introduces its four-terminal multiuser operating system, soon to be released to computer stores. Until then, you can purchase at least one example of each of the major languages to run on their single-terminal DOS 69 operating system.

SWTP S/09

The S/09 has continued as SWTP's most successful product since INTERFACE AGE reported on it last July. This month we'll highlight their System D, a remarkably complete multiuser business computer system.

The System D is equipped with dual full-size dual-density floppy disk drives that yield an impressive 2.5 megabytes of online storage. But even that is simply an auxiliary to the main bitbucket: a 16-megabyte Winchester hard disk unit. Equally generous is the standard complement of 128 kilobytes of main memory, which is allocated among the three supplied CRT display terminals. The terminals are SWTP's

own CT-82 design, one that definitely falls into the "smart" category, even though the screen is limited to but 1640 characters if lower case display is required.

All S/09 systems can be purchased with a good selection of programming languages and aids, including what is billed as the fastest BASIC ever for the 6809 chip. Multiterminal operation is limited to the BASIC environment, a restriction shared by Cromemco and some others in a list.

This product presents a good example of why you shouldn't select a computer by simply running your finger down the price column. Even though $12,000 may sound like a stiff tariff, it includes many essential items that must be purchased as extras on many of the systems under review.

6502 SERIES OF EIGHT-BIT COMPUTERS

Computhink MINIMAX

The MINIMAX is a fresh computer design from a company that gained its reputation as a supplier of floppy disk drives — 4000 shipped so far — to Commodore PET users. The MINIMAX II is the larger of the two available versions, offering 2.4 megabytes of storage capability in its dual 8" flexible disk units. The machine is nicely packaged in a large CRT terminal-style enclosure, and includes a remarkable 108 kilobytes of RAM internally. The display screen is able to display high-resolution graphics, and some slick software is included to produce charts, histograms and business forms.

Computhink is proud that it has produced all the software that runs on the MINIMAX's 6502 microprocessor. BASIC and a machine language assembler are standard, with PLM and FIFTH available. FIFTH was described to us as "a Pascal derivative on FORTH." On the applications side, the MINIMAX offers an extensive DBMS subsystem and four of the "big five" accounting packages as standard equipment.

Ohio Scientific C3

Ohio Scientific's C3-C microcomputer system is absolutely unique in that it features not one but all three of the eight-bit micros under discussion: 6502, 6800 and Z80. The 6502 is the most completely supported in terms of Ohio Scientific-supplied software, so we classify it in this group.

The C3-C was one of the first to marry a dual 8" floppy disk drive with a single large-capacity Winchester disk and market it as a package. The example was followed by several others. The C3-C can be fitted with a copious supply of RAM and an optional multiterminal operating system to allow up to eight users to use the system simultaneously. In addition, C3's can be interconnected via a telecommunications network to provide multiprocessing capabilities.

The Ohio Scientific catalog carries an almost embarrassing array of applications programs, including games, personal computing routines and educational packages. Only a few of the listed programs seem appropriate to a serious business environment, however, and they carry serious business prices.

IBM 5110

IBM's eight-bit 5110 computing system does not use any of the microprocessors listed in the previous pages, but an internally-developed proprietary design. We place it here, between the eight- and 16-bitters, because it has some of the characteristics of both: eight-bit performance and a 16-bit price. Actually, that's a bit unfair to this gargantuan company which is mother and father to business computing. Purchasers of IBM products know they are paying a premium to deal with a company that not only makes very few mistakes in designing, building and marketing, but also boasts a stability that is likely to outlast us all.

The 5110 is a floppy-disk based development on the earlier 5100, which utilized proprietary cassette tapes for data storage. Cassette drives are still available for the 5110 for those who have a library of tapes to run on the earlier model. The unit is housed in a neat tabletop enclosure looking a lot like a large scientific electronic calculator. The small size of the main computer is offset by a large floor-standing rack-style cabinet needed to house the dual 8" floppy disk drives. A second pair of drives can be fitted (bringing the total capacity up to 4.8 megabytes), but they need their own identical enclosure.

In keeping with its small size, the display screen on the 5110 holds but 16 lines of 64 characters each. Applications programs have to keep this limitation in mind. The machine is the only one we know of in this size range which supports the widely revered IBM-developed APL language.

Packaged solutions to business problems have always been an IBM specialty, so we were not surprised to see a selection of well-executed applications packages. They specialize in routines designed around a type of business (dental, travel agency, etc.) rather than generally applicable ones such as payroll — although the latter can be had as well. Some of the software is rented by the month, rather than licensed or sold outright.

16-BIT MICROCOMPUTERS

Alpha Micro AM-1031

We're moving into the "big iron" now in terms of size and price. Expect a 16-bit computer to offer something special in terms of performance, and our first entry, the Alpha Microsystems AM-1031, is no disappointment. The machine is a wayward child of DEC's LSI-11 microcomputer, is inherently multiuser, multitasking, and multiterminal in its operation, and is equipped with a wider array of development software than any system we will cover.

The firm has nearly doubled its number of installed systems since October 1979 — most of them equipped with a cartridge-type hard disk subsystem. Late last year, the company quietly began to upgrade the AM-100/T variant of its basic AM-100 design. The event deserved more fanfare than it occasioned, because the /T is in a new class in terms of pure number crunching performance. No longer saddled with eight-bit memory accesses, the new CPU talks to a true 16-bit RAM card at up to twice the previous rate. The dynamic RAM itself features three extra bits per byte (for a total of 11) which serve to correct all single-bit read errors "on the fly." Coupled with a 50% hop-up of the basic processor speed, the AM-100/T performed our prime number benchmark nearly twice as quickly as its predecessor.

The AM-1031 is the mid-sized Alpha Micro. Its 10-megabyte Control Data Hawk cartridge disk drive has proven itself to be more popular than both the floppy-based systems and the ones equipped with the 90-megabyte Phoenix drive. A single system can be fitted with any combination of these data storage devices, offering more flexibility than a user will probably ever need.

To programmers, the AM-1031's most appealing feature is the nearly 200 languages and utility programs that are standard. BASIC is the primary applications language, and Pascal, LISP and a macro assembler are included. AlphaBASIC is a semi-compiling language that leans towards business applications with its COBOL-like data structures and built-in ISAM capability. Alpha Micro also includes a really slick screen-oriented text editor that can be used for word processing applications as well as program entry. As far as applications programs go, Alpha Micro dealers can serve up a factory-supported package that incorporates general ledger, accounts payable, accounts receivable, payroll and order entry/inventory control.

Digital Microsystems HEX29

The heart of the HEX29 is not really a microprocessor at all, but a brace of four AMD2900 bit-slice chips that team up to provide a minicomputer-like broadside of 16-bit power. The HEX29 fairly cooks, and this capability is further enhanced by its version of Pascal, which compiles programs down into the machine language of the CPU itself, not in an intermediate P-code as others do (see INTERFACE AGE, Jan. 1980, for an in-depth look at the HEX29).

If you can type, you can handle your accounting, word processing and much more on ISC's Small Business Computer!

The Intecolor® 8963 is just one of a complete line of ISC desktop computers designed for businessmen who don't know how— or don't have time— to write programs. It's CP/M® compatible, so you can choose from hundreds of CP/M business programs— programs that have been proven in hundreds of actual applications.

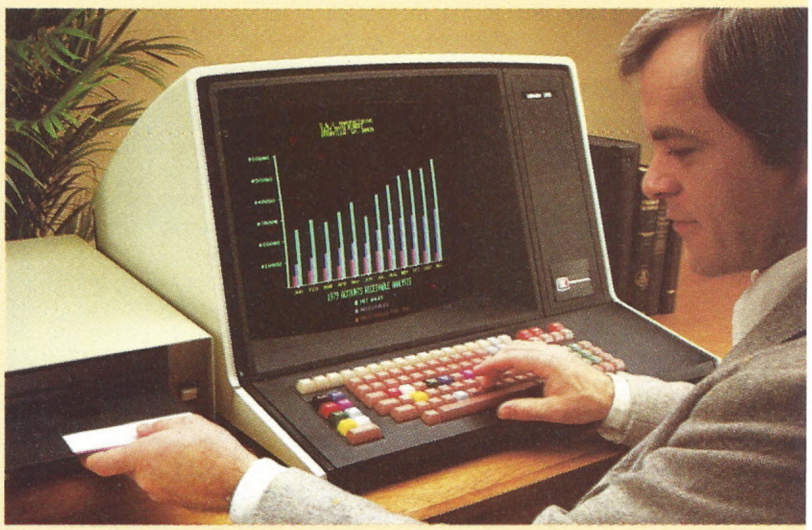
Programs like General Ledger, Accounts Receivable and Payable, Payroll, Mailing List and Inventory Control are now available *in color.* You'll comprehend data faster, thanks to the proven readability of ISC's eight-color display.

To compose letters and other documents quickly and easily, you'll want ISC's unique *color-coded* Word Processing program. With an optional printer, you can print out as many mistake-free originals as you want!

The Intecolor 8963 is just $6395,* and includes a 19" color graphics display, typewriter-like keyboard, dual disk drive for data storage— even a color version of Microsoft® Business BASIC for those of you who *do* want to program.

Don't let your business get behind the times. Call your ISC sales representative or visit your nearest Factory Authorized ISC dealer and get a "hands on" demonstration *today.*

Color Communicates Better



* U.S. domestic price. Unretouched photo of screen. CP/M is a registered trademark of Digital Research Corp.

OEM Quantity Discounts are available to Qualified Dealers and Volume Users of 25 Systems (or greater) per year. Call your nearest ISC Representative listed below.

ISC SALES REPRESENTATIVES: AL: 205/883-8660, **AZ:** 602/994-5400, **AR:** (TX) 214/661-9633, **CA:** Alhambra 213/281-2280, Goleta 805/964-8751, Irvine 714/557-4460, Los Angeles 213/476-1241, Los Altos 415/948-4563, San Diego 714/292-8525, **CO:** 303/355-2363, **CT:** 203/624-7800, **DE:** (PA) 215/542-9876, **DC:** (VA) 703/569-1502, **FL:** Ft. Lauderdale 305/776-4800, Melbourne 305/723-0766, Orlando 305/425-5505, Tallahassee 904/878-6642, **GA:** Atlanta 404/455-1035, **HI:** 808/524-8633, **ID:** (UT) 801/292-8145, **IL:** (No.) 312/564-5440, (So. MO) 816/765-3337, **IN:** (IL) 312/564-5440, **IA:** (Scott County Only) 312/564-5440, (MO) 816/765-3337, **KS:** (MO) 816/765-3337, **KY:** 606/273-3771, **LA:** 504/626-9701, **ME:** (MA) 617/729-5770, **MD:** (VA) 703/569-1502, **MA:** 617/729-5770, **MI:** Brighton 313/227-7067, Grand Rapids 616/393-9839, **MN:** 612/645-5816, **MS:** (AL) 205/883-8660, **MO:** 816/765-3337, **MT:** (CO) 303/355-2363, **NB:** (MO) 816/765-3337, **NH:** (MA) 617/729-5770, **NJ:** (No.) 201/224-6911, (So.) 215/542-9876, **NV:** (AZ) 602/994-5400, **NM:** 505/292-1212, **NY:** Metro/LI(NJ) 201/224-6911, N. Syracuse 315/699-2651, Fairport 716/223-4490, Utica 315/732-1801, **NC:** 919/682-2383, **ND:** (MN) 612/645-5816, **OH:** Cleveland 216/398-0506, Dayton 513/435-7684, **OK:** (TX) 214/661-9633, **OR:** 503/644-5900, **PA:** (E) 215/542-9876, (W) 412/922-5110, **RI:** (MA) 617/729-5770, **SC:** 803/798-8070, **SD:** (MN) 612/645-5816, **TN:** 615/482-5761, **TX:** Austin 512/454-3579, Dallas 214/661-9633, El Paso Area (Las Cruces, NM) 505/524-9693, Houston Only 713/681-0200, **UT:** 801/292-8145, **VT:** (MA) 617/729-5770, **VA:** 703/569-1502, **WA:** 206/455-9180, **WV:** (PA) 412/922-5110, **WI:** (IL) 312/564-5440, **WY:** (CO) 303/355-2363,
EUROPEAN EXPORT SALES: EUROPE: (MA) 617/661-9424, **BELGIUM:** Brussels 02-242-36-04, **DENMARK:** 02-913255, **FRANCE:** Rueil Malmaison 749-47-65, Paris 33-1-306-4606, **GREECE:** Athens 642-1368, **ITALY:** Milano 02600733, **THE NETHERLANDS:** Poeldijk 01749-47640, Amsterdam 020-360904, **SPAIN:** Barcelona 204-17-43, **SWEDEN:** Vallingby 08-380-370, **SWITZERLAND:** Mutschellen 057-546-55, **UNITED KINGDOM:** Bournemouth 0201671181, **WEST GERMANY:** Koblenz 01149-31025/6, **AUSTRALIA & NEW ZEALAND:** Auckland 876-570, Canberra 58-1811, Chermside 59-6436, Christchurch 796-210, Melbourne 03-543-2077, Sydney 02-808-1444, Wellington 644-585, **CANADA:** Dorval 514/636-9774, Ottawa 613/224-1391, Toronto 416/787-1208, Vancouver 604/684-8625, **CENTRAL AND SOUTH AMERICA & CARIBBEAN:** (GA) 404/394-9603, **MEXICO:** Monterrey 564-876, **FAR EAST:** (CA) 213/382-1107, **HONG KONG:** 5-742211, **JAPAN:** (Tokyo) (03) 463-9921, **TAIWAN:** (Taipei) 02-7026284, **MIDDLE EAST:** (GA) 404/581-0243, **EGYPT:** 809933, **ISRAEL:** Ramat Gan 03725749, **KUWAIT:** Kuwait 438-180/1/2, **LEBANON:** Beirut 221731/260110, **SAUDI ARABIA:** Jeddah 27790, Riyadh 25083-39732.
For sales and service in other countries contact ISC headquarters in Norcross, GA., U.S.A.

ISC **Intelligent Systems Corp.®**

Intecolor Drive □ 225 Technology Park/Atlanta □ Norcross, GA 30092 □ Telephone 404/449-5961 □ TWX 810-766-1581

CIRCLE INQUIRY NO. 38

In brief, this machine can hold up to two megabytes of error-correcting Random-Access Memory to be utilized by up to 32 different users simultaneously. Only single-sided, single-density floppy disk drives can be fitted, but Digital Microsystems makes up for that in part by allowing more drives than one will probably use. Actually, the optional 28-megabyte hard disk drive is a "must" for business applications with this powerful system.

The standard BASIC interpreter includes ISAM capability; FORTRAN is also available. The supplied line-oriented text editor rates only fair in the ease-of-use category when compared to the marvelous screen-oriented designs that are becoming commonplace. The HEX29 has no factory-supported applications programs.

Rexon RX30

Rexon stands apart from most of the vendors in this review due to its IBM-like approach to selling computer systems. The RX30 is a machine that you buy as a whole, including installation, maintenance support and, most likely, an integrated set of applications programs assembled by the Rexon dealer. This marketing approach is, of course, older than the microprocessors themselves, but is often accompanied by serious compromises in the performance of the computer itself. Not so with the RX30, in large part because it is constructed around the state-of-the-art 16-bit 8086 micro. It's the first such implementation we've seen, and it looks to be the first of many such applications for this powerful chip.

A 20-megabyte cartridge-type hard disk drive is the only mass storage device available on the RX30. Floppy disks were thought to be inappropriate to this class of machine. That's powerful stuff, coming as it does from Dr. Wang, Rexon's president and single-handed developer of much of today's floppy drive technology.

The RX30 is a BASIC-only computer in the mold of Basic/Four. It's far simpler for a programmer to comprehend than, say, the complex structure of the Alpha Micro computer. The limitations in flexibility inherent with this approach are somewhat offset by the ease of training and programming debugging. The most spectacular item in Rexon's software lineup is IDOL, a combination DBMS and programmer's aid. It's conceivable that one could make a career of writing business applications without ever learning to program in BASIC; simply by using the capabilities of IDOL alone. Most of the RX30's applications software comes via the factory-sponsored dealer Software Exchange. The catalog has over 100 pages and is growing steadily.

Technico SS-16

The SS-16 computer is the only one in our list that utilizes the Texas Instruments' TMS9900 16-bit microprocessor chip. Technico has a good thing all to itself so long as their competitors continue to shun this extremely capable device. Technico has its roots in the industrial process control field, and is only recently making its product available in business garb. The TAS-MU-DFD is an SS-16 with 64 kilobytes of RAM and a dual 8" floppy disk drive housing a half-megabyte of storage capacity. Winchester-technology hard disk drives can be added to give up to 40 megabytes more of data storage.

A multiuser operating system, which is standard, allows up to 18 terminals to be connected simultaneously. Actually, there are limitations. Space restrictions in the CPU box hold you to 12 users if 192 kilobytes of RAM are fitted, or six users with 224 kilobytes. It would be hard to call that a serious shortcoming.

Since February, Technico has released packages to handle all of the "big five" business applications, in addition to a date base manager. Although it has traditionally been a bit behind the times in the area of word processing, Technico is about to release a blockbuster. We were sworn to secrecy regarding most of the details, but. . .would you believe — multicolored entry?

Three Rivers Computer PERQ

It is appropriate that our review close with PERQ, since it embodies what might become the future of small business computers. The PERQ (pronounced "perk," not "pur-que") borrows almost nothing from the designs arrayed before you in this issue; it strikes out into territory that was previously explored only in the thought experiments of advanced think-tank operations.

PERQ is intended to be the all-in-one work station of the electronic office of tomorrow. Its powerful computing capabilities are optimized to provide local problem-solving power as well as intercommunication with a shared-resource network consisting of other PERQs and larger systems. It is inherently multitasking, but these tasks are designed to serve the single person who sits at its keyboard. This person is served by a quarter megabyte of RAM and 12 megabytes of hard disk storage as well.

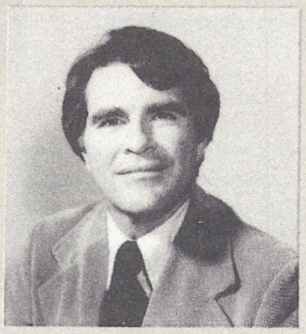
The display terminal is unique. Three Rivers Computer has applied its expertise in high-resolution display technology to serve up a CRT, which can speak in a dozen different type faces, with proportional justification thrown in for good measure. The screen can be divided into several windows, each showing the status of the various multiprocesses that the computer performs. The display is oriented vertically like a sheet of typewriter paper, and is ideally adapted to word processing entry. Actually, the PERQ looks beyond word processing as we know it today to the future when the vast majority of written communication will be transmitted electronically. Why clutter up your office with paper when everyone you correspond with has a PERQ-like machine to receive your communications via satellite signals?

The PERQ also features a Touch Table that translates pencil or finger pressure directly into signals to direct the cursor on the display screen. A speech output module is also standard equipment.

The Three Rivers product utilizes a proprietary 16-bit processor design which has the native language of Pascal P-code. This is an optimized-efficiency approach which is sure to be quite common in 1990s-vintage systems. No other language is available or needed on the PERQ.

The PERQ is definitely in that class of "solutions looking for a problem" machinery. The biggest threat to its probable success would be users who lack the imagination to apply mind-stretching capabilities properly.

The power of the computer systems represented here seems almost commonplace today. Only a few years ago they were misunderstood toys gracing the offices of a few forward-thinking business people. Most of us were left to play catch-up; and some, it is sad to say, remain ignorant of the rewards available to those who would make a home in their businesses for a computer system. □



ABOUT THE AUTHOR

Tom Fox has twenty years experience in the field of electronics, thirteen years in computer systems and their application to business and industry. President of FoxWare Systems Corporation, of Irvine, California, he is also past Director of Engineering at Structured Micro Systems, Inc., where he headed up the development of a computerized telephone answering service and centralized alarm monitoring system. He can be reached at 17925-G Sky Park Circle, Irvine, CA 92714, telephone (714) 957-9331.

PEARL™!

Skeptics are still saying a program like PEARL won't be available until the mid-1980's

They're wrong, of course. PEARL is available now.

PEARL (Producing Error-free Automatic Rapid Logic) brings application generation software to microcomputer users at four user levels.

Very simply, PEARL means that anyone with a microcomputer with 48K memory can use one of the four PEARL programs to generate new applications software with a minimum of time, trouble and training.

PEARL runs under CP/M* and is designed as a powerful menu-driven tool. PEARL Level 1 provides non-programmers with the capability to program their own new applications automatically. Users respond to on-screen prompts as they define their system; no programming knowledge is required.

PEARL Levels 2 and 3, for experienced programmers, provide sophisticated processing for complex applications. In just fractions of the time it used to take to design, code, and test a program, PEARL will generate and compile a new application. PEARL Level 4, the most advanced of the PEARL packages, provides programmers with the capability to include many features normally associated with customized turnkey systems.

Introductory prices. Trade-in allowances for upgrade to higher levels.

PEARL Level 1	For Personal Computing	$ 90	
PEARL Level 2	The Business Assistant	$ 200	
PEARL Level 3	For Software Developers	$ 400	May 1980
PEARL Level 4	For Turnkey Systems Developers	$ 800	June 1980

evolutionary software for the 1980's

Computer Pathways Unlimited, Inc.

Corporate Office: 2151 Davcor Street S.E.
Salem, Oregon 97302
503-363-8929

CIRCLE INQUIRY NO. 13

THE COMPARISON TABLES EXPLAINED

By Tom Fox
Systems Editor

Like the smorgasbord at a Swedish wedding reception, the tables on the following pages are fairly laden with tasty morsels that are the visible fruits of a small army of behind-the-scenes artists. The creators of the masterpieces arrayed before you do not practice their skills in the kitchen, but in the development labs that are the most guarded corners of the many computer factories represented here.

The six tables of data can help in comparing various aspects of the systems we are looking at this month. Most of the information is self-explanatory, but some of the categories can be a little tricky. This is mainly due to the fact that tabular headings that do a good job of describing one manufacturer's system are often inadequate to describe another. Each system has unique properties not shared by any of the others. We did the best we could, and personally checked with each manufacturer to make sure all of the entries are up-to-date. Keep in mind as you read: "n/a" means "information not available" or, in some cases, "refused."

TABLE 1 — HARDWARE DATA

This chart carries general information about the hardware elements of the system. Most importantly, it lists a model number, price, and tells what major components are included in that price. In this and the following tables, the indication "Std" means the hardware or software item is included as standard equipment at the quoted price. "Opt" indicates the item is optional at extra cost. If the extra dollar amount was made available, we included it.

Under "ENCLOSURE," we indicate if the computer is supplied in a free-standing tabletop housing, is integrated into a desk-style work station, is equipped for rack mount-

ing, or a combination of these. The "BUS" column tells us if the plug-in-cards that make up the computer conform to a known industry standard. You can, in theory, interchange different manufacturers' circuit boards if they both conform to the same bus protocol. This is done every day, but be warned that the maneuver is sometimes doomed because of minute discrepancies among the various designs.

"RAM" stands for random access memory, the fast semiconductor memory element utilized by all of the systems we are reviewing this month. The amounts are shown in kilobytes, or thousands of bytes. (A byte is equivalent to the amount of storage required to remember a single character, such as "A".) All of the systems we are looking at are equipped with one or more disk drives, either of the flexible diskette ("floppy") variety, or the higher-performance hard disk drive. Some systems come complete with both; the floppy drive often included as a means to create backup copies of data on Winchester-technology hard disk drives. Other than the obvious advantage of having more storage space, hard disk drives are commonly several times faster than the floppy variety.

In Table 1 is an indication as to whether or not a cathode-ray tube (CRT) display terminal or hard copy printer is included in the basic price. Several of the manufacturers do not provide these devices at all; they let the local dealers buy them separately and integrate them into a system before delivering the assemblage to the ultimate user. The peripheral devices can form a significant fraction of the purchase price, so take that into account when comparing system costs.

TABLE 2 — DISK DRIVE DATA

Here we take a closer look at that most critical element of a computer system: the mass storage device. Shown is the number of drives supplied as standard, and the maximum size that can be fitted in the future. Capacities are shown in kilobytes for the floppy disk drives; megabytes (millions of bytes) in the hard disk case. Remember that 1,000 kilobytes equal one megabyte.

All floppy disk drives are built to allow you to remove the diskette itself and store it away for safekeeping or to maintain a library of programs, data, etc. It's of critical importance that at least one extra copy of each diskette be maintained as well. There are several kinds of human and machine errors that can cause the data on a diskette to be destroyed with alarming ease. The activity of creating the backup copy requires that at least two diskette drives be fitted to the computer system, the master being copied from and the "scratch" disk being copied onto.

The same principles, of course, apply to hard disk drives. In this case, however, the amount of data that can accidentally be destroyed is much greater. Hard disks listed as the "cartridge" type in Table 2 contain a certain amount (usually half) of their storage on an internal, non-removable disk platter. The cartridge itself is a removable disk that can be inserted into the drive to back up the data contained on the non-removable one. Backup copies of critical data are thus nearly as easy to make as the floppy disk systems. Winchester-technology hard disks are fundamentally different, in that all data is contained on a non-removable disk platter. Winchester devices contain their spinning disk within a hermetically sealed environment, and any data copies must be made onto some other external device; in some cases, floppy disk drives; in others, special magnetic tape units. Winchester technology offers today's best bargain in bytes-per-buck storage, but the world is still waiting for someone to solve the knotty problem of backing up the data in a convenient, economical manner. Many simply skip the backup process — an invitation to disaster, in our opinion.

It may be nit-picking, but we've noticed that computer salesmen are beginning to list the "unformatted" capacity of their disk drives rather than the "formatted" or usable quantity. A certain portion of each disk or floppy diskette is occupied with "housekeeping" data, so all of that space isn't really available for your own use. (An example: One manufacturer's implementation of the popular Shugart SA-4000 Winchester drive requires 6 megabytes of the disk's 29-megabyte capacity for internal formatting information, leaving but 23 megabytes for your use.) We have tried mightily to list only the formatted capacity in Table 2, but in a few cases, even the factory representatives didn't know!

TABLE 3 — PERIPHERALS DATA

In this chart, we have gathered together the last of the hardware-related information. Each of the systems utilizes some form of a CRT display terminal for primary data input and output, although some are priced separately. A few have a graphics capability in addition to the basic ability to display letters and numbers. The MAXIMUM QUANTITY column indicates how many terminals can be connected to the system simultaneously in a multi-user environment.

A printer is an essential part of any business computer system. Two of the columns in Table 3 briefly describe those available. Matrix printers form the characters on the page as a collection of tiny dots. They are usually faster, less expensive and more reliable than character printers, but are not suitable for word-processing applications if you want the final result to appear as if it were produced on a typewriter. Fully formed character printers make their image like a typewriter, with a single stroke. Nearly all character printers are of the spinning daisywheel or thimble variety, which represent a real performance advancement over the earlier tumbling golf ball technology of IBM's Selectric. Printer speeds are shown as the average number of characters produced per second (cps) or, in some cases, the number of complete lines of text printed per minute (lpm). Don't be concerned if the chart comes up "none" in both the STANDARD and OPTIONAL categories. This just means that the local dealer must buy directly from the printer manufacturer, the same as the computer manufacturer.

The final column in Table 3 is a catch-all for the special and unique hardware attachments that form a large part of a computer's personality. We didn't list the serial and/or parallel interfacing port options.

TABLE 4 — SYSTEM SOFTWARE DATA

System software includes programs normally provided by the manufacturer to manage the central function of moving data around among the computer memory, disk, printer, and terminal(s). It also includes software than can be utilized by programming personnel to create usable application programs (listed on the next chart). We have made a special effort to research the prices for software, to highlight the significant cost that can be represented by this category of investment. If having a large library of purchasable programs is important in your business, pick a system that can run with one of the widely used operating systems. CP/M leads the pack in this race, but several others follow closely.

Column headings are included for the four most popular programming languages (BASIC, COBOL, FORTRAN and Pascal) as well as the assembler which can produce runnable programs in the native language

Table 1. Hardware Data

MANUFACTURER	SYSTEM	BASIC PRICE	ENCLOSURE			CPU		BUS	RAM		STANDARD PERIPHERALS		
			TABLETOP	DESK	RACK	TYPE	BITS		STD	MAX	DISK DRIVE	TERMINAL	PRINTER
Alpha Micro	AM-1031	$17,835	Std	No	Opt	WD16	16	S-100	64K	1024K	1 Hard	No	No
Altos	ACS8000-6/MU4	$11,960	Std	No	No	Z80	8	None	208K	208K	2 Floppy + 1 Hard	No	No
Computhink	MINIMAX II	$ 9,200	Std	No	No	6502	8	None	108K	108K	2 Floppy	Yes	No
Cromemco	System 3	$ 6,990	Opt	Opt	Std	Z80	8	S-100	64K	512K	2 Floppy	No	No
Digital Microsystems	HEX29	$15,900	Std	No	No	2900	16	Prop.	256K	1000K	2 Floppy	No	No
Heath	WH89-CS	$ 2,895	Std	No	No	Z80	8	None	48K	48K	1 Floppy	Yes	No
IBM	5110	$19,475	Std	No	Std	Prop.	8	Prop.	32K	64K	2 Floppy	Yes	No
IMS	Series 8000	$ 4,500	Std	Opt	Opt	Z80	8	S-100	32K	256K	2 Floppy	No	No
Intertec	SuperBrain	$ 2,995	Std	No	No	Z80	8	None	32K	64K	2 Floppy	Yes	No
Micro V	MicroStar I	n/a	Std	Opt	No	8085	8	None	64K	64K	2 Floppy	No	No
MicroDaSys	millie	$ 3,999	Std	Opt	No	Z80	8	S-100	48K	60K	1 Floppy	Yes	No
North Star	HRZ-2-64K-D	$ 5,820	Std	No	No	Z80	8	S-100	64K	64K	2 Floppy	Yes	Yes
Ohio Scientific	C3-C	$ 9,900	No	No	Std	6502 Z80 6800	8 8 8	Prop.	48K	384K	2 Floppy + 1 Hard	No	No
Pertec	PCC 2000	$ 9,995	Std	No	No	8085	8	P-100	64K	64K	2 Floppy	Yes	No
Radio Shack	TRS-80 Model II	$ 3,899	Std	Opt	No	Z80	8	Prop.	64K	64K	1 Floppy	Yes	No
Rexon	RX30	$32,500	No	No	Std	8086	16	Prop.	64K	128K	1 Hard	Yes	Yes
Smoke Signal	9822	$ 4,344	Std	Opt	Opt	6809	8	SS-50	48K	192K	2 Floppy	No	No
SWTP	System D	$12,000	Opt	Std	No	6809	8	SS-50	128K	768K	2 Floppy + 1 Hard	Yes (3)	No
Technico	TAS-MU-DFD	$ 7,895	Std	No	Opt	9900	16	Prop.	64K	224K	2 Floppy	No	No
TEI	3400	n/a	No	Std	No	8085	8	S-100	64K	64K	2 Floppy	Yes	No
Three Rivers Computer	PERQ	$19,500	Std	No	Std	Prop.	16	Prop.	256K	1000K	1 Hard	Yes	No
Vector Graphic	System B	$ 5,463	Std	Opt	No	Z80	8	S-100	56K	56K	2 Floppy	Yes	No
Zilog	MCZ-1/70-2	$19,835	No	No	Std	Z80	8	Z-bus	64K	64K	1 Hard	Yes	No

Table 2. Disk Drive Data

MANUFACTURER	SYSTEM	NUMBER OF DRIVES/TOTAL CAPACITY (bytes) FLOPPY DISKETTE SIZE	STD	MAXIMUM	HARD DISK TYPE	STD	MAXIMUM
Alpha Micro	AM-1031	8"	None	8/9600K	Cartridge	1/10M	4/360M
Altos	ACS8000-6/MU4	8"	2/1000K	4/8000K	Winchester	1/14M	2/58 M
Computhink	MINIMAX II	8"	2/2400K	2/4800K	—	None	—
Cromemco	System 3	8"	2/1024K	4/2048K	Winchester	None	4/40 M
Digital Microsystems	HEX29	8"	2/512K	8/2048K	Winchester	None	2/56 M
Heath	WH89-CS	5¼"	1/100K	1/100K	—	None	—
IBM	5110	8"	2/2400K	4/4800K	—	None	—
IMS	Series 8000	8"	2/486K	4/3856K	Cartridge	None	2/180M
Intertec	SuperBrain	5¼"	2/265K	2/700K	Winchester	None	4/72 M
Micro V	MicroStar I	8"	2/2000K	4/4000K	Winchester	None	1/20 M
MicroDaSys	millie	8"	1/500K	4/2000K	Cartridge	None	4/90 M
North Star	HRZ-2-64K-D	5¼"	2/360K	4/1440K	Winchester	None	4/72 M
Ohio Scientific	C3-C	8"	2/544K	2/1088K	Winchester	1/23M	1/23 M
Pertec	PCC 2000	8"	2/1200K	2/1200K	Cartridge	None	4/80 M
Radio Shack	TRS-80 Model II	8"	1/486K	4/1944K	—	None	—
Rexon	RX30	—	None	—	Cartridge	1/20M	2/40 M
Smoke Signal	9822	8"	2/2000K	8/7500K	Cartridge	None	1/20 M
SWTP	System D	8"	2/2500K	4/5000K	Winchester	None	1/16 M
Technico	TAS-MU-DFD	8"	2/512K	4/1024K	Winchester	None	4/40 M
TEI	3400	8"	2/1986K	8/7944K	Winchester	None	32/442M
Three Rivers Computer	PERQ	8"	None	1/1024K	Winchester	1/12M	1/24 M
Vector Graphic	System B	5¼"	2/630K	4/1260K	—	None	—
Zilog	MCZ-1/70-2	8"	None	4/1200K	Cartridge	1/10M	4/40 M

of the microprocessor itself. Under OTHERS, we list additional programming languages and major utilities that ease the applications programmer's task. In addition, all of the systems make available some kind of a text-entry capability for keying in the source programs in the various languages.

TABLE 5 — APPLICATIONS SOFTWARE DATA

Taken as a whole, writing applications programs for computers is a massive task, occupying a rapidly increasing fraction of the world's workforce. There is a snowballing trend for computer manufacturers to deliver their machines with useful programs that can be put to use right away. Their intent is to apply some standardization to programs which are run on their products, and to remove a significant barrier to their sales. The effort is ambitious and well-motivated. To be successful, the programs themselves must be very nearly perfect and generally applicable in a wide variety of applications situations.

Table 5 lists the major applications packages that are available from the various manufacturers. The "big five" are there (general ledger, accounts payable, accounts receivable, payroll and order entry/inventory control), as are a pair of applications that are becoming featured by more and more computers these days: word processing and data base management system (DBMS).

Modern word processing programs consist of two parts: a means to enter textual data from a CRT terminal, and a way to transform the information and send it to a hard-copy printer. Many of the word processors feature a "screen-oriented" editor for data entry which is rapidly becoming today's state-of-the-art.

First-time computer users who have never uttered the acronym DBMS often find it soon becomes the most used program in the machine. Applied intelligently, a good DBMS can organize everything from the annual report to your daily appointment calendar. There is little standardization in DBMS programs, so make sure the one you purchase is appropriate to the size and complexity of your needs.

TABLE 6 — CORPORATE DATA

The final chart in the series zeroes in on the companies that make the computer systems being reviewed in this issue. Included are size, age, and retail outlet base, as well as the manufacturing history of the system itself. GROSS SALES are for the most recent complete business year and include the receipts of any parent company. Also shown are the current number of retail outlets and service centers, if different.

The final two columns tell the birthdate and total number delivered of the system being reviewed in this issue. In a few cases, we have allowed the manufacturer to include nearly identical predecessors of the reviewed system if the differences are small and evolutionary.

We will let you roam through the tables like the guest at a Swedish wedding reception, comparing the relative merits of each dish and selecting the one that suits your taste. We make no recommendations as to which might be best, as each and every one of them will be the optimum choice given a particular business situation. Assembling this mass of data was the easy part. The difficult task, now, is yours: choosing the system that enables your business to run more smoothly and profitably.□

Table 3. Peripherals Data

MANUFACTURER	SYSTEM	CRT DISPLAY TERMINAL				PRINTER TYPE/SPEED		OTHER
		PACKAGE	GRAPHICS	CHARACTERS	MAX QTY	STANDARD	OPTIONAL	
Alpha Micro	AM-1031	Separate	No	1920	12+	None	None	Real-time Clock (Std) Magnetic Tape ASYNCH/SYNCH Communications
Altos	ACS8000-6/MU4	Separate	No	1920	4	None	None	Magnetic Tape Floating Point
Computhink	MINIMAX II	Integrated	Std	1920	1	None	Matrix/60 cps Matrix/150 cps	None
Cromemco	System 3	Separate	No	1920	7	None	Matrix/60 cps Matrix/180 cps Character/55 cps	PROM Programmer Color Graphics A/D & D/A
Digital Microsystems	HEX29	Separate	No	1920	32	None	Matrix/150 cps Matrix/300 cps	None
Heath	WH89-CS	Integrated	No	1920	1	None	Matrix/40 cps Matrix/150 cps Character/45 cps	None
IBM	5110	Integrated	No	1024	1	None	Matrix/80 cps Matrix/120 cps	Mag Tape Cartridge BISYNCH Comm. ASYNCH Comm.
IMS	Series 8000	Separate	No	1920	8	None	None	None
Intertec	SuperBrain	Integrated	No	2000	1	None	None	S-100 Bus Adapter
Micro V	MicroStar I	Separate	No	1920	2	None	Matrix/112 cps Matrix/300 lpm Character/55 cps	None
MicroDaSys	millie	Separate	Opt	2000	1	None	Character/55 cps	Color Graphics
North Star	HRZ-2-64K-D	Separate	No	1920	1	Matrix/112 cps	Character/55 cps	Floating Point
Ohio Scientific	C3-C	Separate	No	1920	8	None	Matrix/110 cps Matrix/125 lpm Character/55 cps	Networking B/W Graphics
Pertec	PCC 2000	Integrated	Std	1920	5	None	Matrix/120 cps Character/55 cps	None
Radio Shack	TRS-80 Model II	Integrated	Std	1920	1	None	Matrix/60 cps Matrix/120 cps Character/50 cps	None
Rexon	RX30	Separate	No	1920	8	Matrix/150 cps	Matrix/300 lpm	None
Smoke Signal	9822	Separate	No	1920	1	None	Matrix/165 cps	B/W Graphics
SWTP	System D	Separate	Std	2024	16	None	Matrix/60 lpm Matrix/120 cps Character/45 cps	None
Technico	TAS-MU-DFD	Separate	No	1920	18	None	Matrix/180 cps	A/D & D/A RAM Battery
TEI	3400	Separate	No	2000	1	None	Matrix/150 cps	None
Three Rivers Computer	PERQ	Integrated	Std	5490	1	None	Matrix/300 lpm Character/45 cps	Touch Tablet (Std) Speech Output (Std) GPIB Interface (Std) Networking
Vector Graphic	System B	Separate	No	1920	5	None	Matrix/150 cps Character/55 cps	A/D & D/A B/W Graphics Video Digitizer
Zilog	MCZ-1/70-2	Separate	No	1920	5	None	Matrix/140 cps Character/55 cps	None

Table 4. Systems Software Data

MANUFACTURER	SYSTEM	OPERATING SYSTEM		PROGRAMMING LANGUAGES					
		SINGLE-USER	MULTI-USER	ASSEMBLER	BASIC	COBOL	FORTRAN	PASCAL	OTHERS
Alpha Micro	AM-1031	None	AMOS	Std	Std	No	No	Std	LISP (Std) ISAM (Std) SORT (Std)
Altos	ACS8000-6/MU4	CP/M $150	AMEX $600	$100	$150	$750	$500	$200	OASIS $500 APL $500 KSAM $450
Computhink	MINIMAX II	DOS	None	Std	Std	No	No	No	PLM FIFTH
Cromemco	System 3	CDOS	Multi-user BASIC $800	$95	$95	$95	$95	No	RATFOR $195 TRACE $95 Struct. BASIC $295
Digital Microsystems	HEX29	None	HOST	Std	Std	No	Opt	Opt	ISAM
Heath	WH89-CS	H-DOS $100	None	Std	Std	No	No	No	Microsoft BASIC $100
IBM	5110	n/a	None	No	Std	No	No	No	APL $1000 Sort Utility Subroutine Library
IMS	Series 8000	CP/M	FAMOS $1500	Opt	$100	Opt	Opt	Opt	OMNIX $350 CAP-CPP CBASIC $500
Intertec	SuperBrain	CP/M	None	Std	$350	No	$500	No	None
Micro V	MicroStar I	None	StarDOS	Opt	Std	Opt	Opt	Opt	CP/M CAP-CPP
MicroDaSys	millie	CP/M	None	Std	Std	$600	$400	$350	CBASIC $100 Microsoft BASIC $300
North Star	HRZ-2-64K-D	NS/DOS	None	No	Std	No	No	$199	None
Ohio Scientific	C3-C	OS-65/U	Level 3 $400	Std	Opt	Opt	Opt	Opt	OS-CP/M $600
Pertec	PCC 2000	DOS	MTX	No	Std	Opt	Opt	No	CP/M
Radio Shack	TRS-80 Model II	TRSDOS	None	No	Std	No	No	No	None
Rexon	RX30	None	RECAP	No	Std	No	No	No	IDOL
Smoke Signal	9822	DOS 69	None	$40	$100	Opt	$150	$250	Compiler BASIC $325
SWTP	System D	FLEX-09	Multi-user BASIC $150	$40	$65	No	No	$250	DEBUG $75 Sort/Merge $75 PILOT $250
Technico	TAS-MU-DFD	None	MU/OS	Std	Std	No	$990	No	IIA (Std)
TEI	3400	TDOS	None	Opt	Opt	Opt	Opt	No	SORT (Std)
Three Rivers Computer	PERQ	None	DOS	No	No	No	No	Std	Symbolic Debugger Screen Window Mgr.
Vector Graphic	System B	CP/M	Timeshare Monitor $250	Std	Std	$500	No	No	UNIVIS (Std) APL $400 ASYNCH Comm. $150
Zilog	MCZ-1/70-2	None	RIO	Std	$500	$850	$950	$950	PLZ $500 ASYNCH Comm. $500

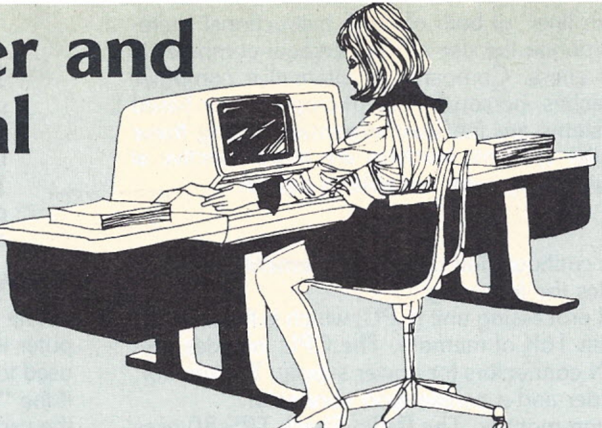
Table 5. Applications Software Data

MANUFACTURER	SYSTEM	WORD PROCESSING	DBMS	G/L	A/P	A/R	PAYROLL	INVENT. CONTROL	OTHERS
Alpha Micro	AM-1031	Std	No	Opt	Opt	Opt	Opt	Opt	None
Altos	ACS8000-6/MU4	$500	No	No	No	No	No	No	None
Computhink	MINIMAX II	No	Std	Std	Std	Std	Std	No	None
Cromemco	System 3	$95	$95	No	No	No	No	No	Multi-user DBMS $195
Digital Microsystems	HEX29	No	No	No	No	No	No	No	None
Heath	WH89-CS	$495	No	No	No	No	No	No	None
IBM	5110	No	No	$75/mo	$75/mo	No	$80/mo	No	Travel Agency Accounting Mortgage Closing Client Accounting Dental Accounting
IMS	Series 8000	$150	Opt	Opt	Opt	Opt	Opt	Opt	WordStar $495 Apartment Management
Intertec	SuperBrain	No	No	No	No	No	No	No	None
Micro V	MicroStar I	Opt	Std	Opt	Opt	Opt	Opt	Opt	Mail List
MicroDaSys	millie	$275	$125	$99	$99	$99	$99	No	WordStar $495 Medical Billing $895 Pencil Sharpener $195 Star Brightener $150
North Star	HRZ-2-64K-D	No	No	No	No	No	No	No	None
Ohio Scientific	C3-C	$200	$300	$300	$300	$300	$300	$300	Educational Pkg. Purchasing Estimating/Quotation Bill of Materials Mail List
Pertec	PCC 2000	No	No	Opt	Opt	Opt	Opt	Opt	None
Radio Shack	TRS-80 Model II	No	No	$199	No	$299	$399	$199	Mail List $79
Rexon	RX30	No	Std	No	No	No	No	No	None
Smoke Signal	9822	$80	No	Opt	Opt	Opt	Opt	Opt	None
SWTP	System D	$35	No	$595	$600	$600	No	$100	Mail List $50 Shipping/Receiving $75
Technico	TAS-MU-DFD	Opt	Std	$2000	Opt	Opt	$1000	Opt	None
TEI	3400	No	Opt	No	No	No	No	No	None
Three Rivers Computer	PERQ	Std	No	No	No	No	No	No	None
Vector Graphic	System B	$450	$350	Std	Std	Std	Std	Std	None
Zilog	MCZ-1/70-2	No	No	No	No	No	No	No	None

Table 6. Corporate Data

MANUFACTURER'S ADDRESS	GROSS SALES	ESTAB.	DEALERS		SERVICE CENTERS		REVIEWED SYSTEM	
			US	FOREIGN	US	FOREIGN	FIRST DELIVERY	TOTAL QTY
Alpha Microsystems 17881 Sky Park North Irvine, CA 92714	$17M	1977	170	30	10	30	Apr 1977	4000
Altos Computer Systems 2360 Bering Drive San Jose, CA 95131	$10M	1977	100	300	100	300	Jan 1975	5000
Computhink 965 W. Maude Avenue Sunnyvale, CA 94086	n/a	1978	25	75	25	75	Sep 1979	300
Cromemco, Inc. 280 Bernardo Avenue Mountain View, CA 94040	n/a	1974	107	38	107	38	Feb 1978	2000 +
Digital Microsystems 4448 Piedmont Avenue Oakland, CA 94611	n/a	1975	n/a	n/a	n/a	n/a	Dec 1979	20
The Heath Company Benton Harbor, MI 49022	n/a	1926	55	n/a	55	n/a	Aug 1979	n/a
IBM/General Systems Division 4111 Northside Parkway Atlanta, GA 30301	$2.3B	1924	n/a	n/a	n/a	n/a	n/a	n/a
Industrial Micro Systems 628 N. Eckhoff Street Orange, CA 92688	$4M	1975	75	25	75	25	Jul 1979	1000
Intertec Data Systems 2300 Broad River Road Columbia, SC 29210	$6M	1973	250	30	50	50	Oct 1979	4000
Micro V Corporation 17777S.E. Main Street Irvine, CA 92714	n/a	1978	40	35	40	9	Sep 1978	600
MicroDaSys P.O. Box 36051 Los Angeles, CA 90036	$1M	1977	200	10	5	5	Feb 1979	125
North Star Computers 1440 Fourth Street Berkeley, CA 94710	n/a	1976	200	100	200	100	Dec 1977	10000
Ohio Scientific 1333 S. Chillicothe Road Aurora, OH 44202	n/a	1975	175	25	2	0	Jan 1979	n/a
Pertec Computer Corporation 12910 Culver Boulevard Los Angeles, CA 90066	$148M	1967	100	35	30	35	Jan 1979	1500
Radio Shack 1300 One Tandy Center Fort Worth, TX 76102	$1.2B	n/a	150	0	100	0	Jul 1979	n/a
Rexon Business Machines 5800 Uplander Way Culver City, CA 90230	$1.5M	1978	42	2	13	2	Jul 1979	200
Smoke Signal Broadcasting 31336 Via Colinas Westlake Village, CA 91361	n/a	1976	80	20	40	20	Feb 1980	100
Southwest Technical Products 219 W. Rhapsody San Antonio, TX 78216	n/a	1964	125	30	125	30	Jun 1979	80
Technico, Inc. 9051 Red Branch Road Columbia, MD 21045	$3.4M	1965	14	26	4	4	Dec 1976	6000
TEI, Inc. 5075 S. Loop East Houston, TX 77033	n/a	1967	100	10	100	10	Dec 1979	500
Three Rivers Computer Corporation 160 N. Craig Street Pittsburgh, PA 15213	n/a	1974	0	0	1	0	Mar 1980	30
Vector Graphic, Inc. 31364 Via Colinas Westlake Village, CA 91361	$40M	1976	225	50	225	50	Jan 1979	10000
Zilog, Inc. 10460 Bubb Road Cupertino, CA 95014	$40M	1974	5	3	9	6	Jan 1980	n/a

The Teacher and the Personal Computer:

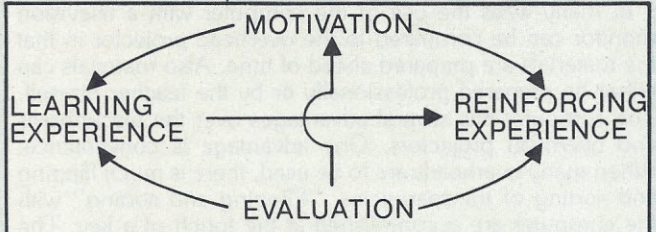


Alternatives in Instruction

By Samuel W. Spero

INTRODUCTION

The instruction process whether it is taking place at the elementary school level or in college must include four elements. These are: motivation, learning experience, reinforcing experience, and evaluation. The diagram below illustrates the relationship between these four elements.



Learning Relationships

While the order in which these elements take place or whether these elements are explicitly or implicitly addressed can vary, in one way or another these four elements must be accounted for in all good teaching.

The manner in which a teacher implements any or all of these four elements is called *instructional strategy.* The instructional strategy, related to the individual teacher's personality and style, should be clearly differentiated from the *curriculum objective.* The curriculum, what the students learn, can be stated very clearly and unambiguously, but the objective does not have to include the manner in which it is to be learned.

Let us review briefly these four elements. A thorough understanding of a teacher's instructional strategies will help clarify the particular strengths of personal computers.

If a student is not motivated, there is little a teacher can do to help the student learn. Punishment is often used to threaten the unmotivated students. In the short range this occasionally works with certain groups, but in making a lasting impression this approach is not recommended. Motivation can also be achieved by presenting the material in an interesting and/or relevant fashion using the student's own intrinsic curiosity. The teacher chooses an instructional strategy which includes *motivational elements* so that students will be interested in what follows.

The actual curriculum objective is introduced in a *learning experience.* The learning experience need not be a lecture, nor is it necessarily several pages in a text. Films, laboratories, field trips are all possible learning experiences. The teacher must decide which learning experience is most appropriate for the particular objective. For example, a lecture is not appropriate for learning to drive, but it might be for learning about history.

Learning theory specialists have discovered that no matter how impressive the learning experience, it must be *reinforced* by other experiences before learning can actually take place. This *reinforcing experience* is in fact the basis for homework. For example, in a mathematics class the students watch the teacher work out a problem in class— the learning experience in this case is lecture and blackboard. Their reinforcing experience is to work on problems from their text, using the same procedure employed by their teacher. The reinforcing experience uses the text and overt responses of the students. Any learning experience can also be a reinforcing experience.

Finally, as part of the teacher's overall instructional strategy, there must be *evaluation.* By evaluation we not only refer to grading students, but also an evaluation of the instructional strategies themselves. The teacher must determine if the instructional strategies were effective. If the students learned, was it because of, or in spite of the instruction strategy; if the students did not learn, where did the instructional strategy fail? Perhaps the students were not motivated or the learning and reinforcing experiences were not appropriate. Evaluation is important to the total instructional process.

When a teacher considers a new medium for incorporation into his or her instructional strategies, many questions must be asked:

Will this medium help motivate my students?
Will it provide a more meaningful learning experience?
Will it create a more relevant reinforcing experience?
Will it permit me to perform a more effective evaluation of my students and my instructional strategies?

TEACHING WITH THE PERSONAL COMPUTER

Microcomputers are making their way into the classroom and becoming one of the most effective instructional strategies an instructor can use.

One teaching strategy uses Radio Shack's TRS-80 to drive a 24-inch television monitor instead of the 12-inch monitor which is normally sold with it. Because of the graphics capability of the TRS-80, as well as its 32-character-per-line output on the television screen, it is possible for all the students to simultaneously view the computer printout on the screen. Using this feature we have been able to develop computer-based instructional units in which the computer — with the television monitor—functions as a "dynamic blackboard." The computer is used to generate material on the television screen which provides a framework for classroom discussion. Used in this way, the personal computer has made the conventional classroom, with its 30 or 40 students, far more exciting and has led to improved learning, according to teachers using this strategy.

The second instructional strategy used is the small, portable, high-speed printer (the Quikprinter I) available with the TRS-80. It generates printed materials for motivation, enrichment, reinforcement as homework assignments or data for a learning experience. It can also be used for evaluation and measurement. The materials generated by the printer can be duplicated for the entire class by the computer itself which prints at 180 characters per second. The computer can also print individualized sets of materials for the students where the computer prints each student's name at the top of their unique problem assignment. The use of the computer to generate classroom materials has led to increased student motivation and learning without substantially increasing the work of the teacher.

At the "bottom line" of both of these instructional strategies which incorporate the use of the personal computer is their cost-effectiveness. Compared to alternative computer hardware approaches, personal — or microprocessor based — computer systems are the least expensive. Using these personal computer systems, teachers are more effective at accomplishing what they were not able to before.

HARDWARE

The hardware configuration used to implement the above strategies includes the following:

a. the central processing unit (CPU) which is the TRS-80 Level II with 16K of memory. The CPU includes keyboard, DIN connectors for power supply, TV monitor, tape recorder and a parallel-port connector.

b. the television monitor. The Radio Shack TRS-80 uses a DIN connector to connect the television monitor to the CPU. The DIN connector is available at any electrical supply store, and the wire connections are straightforward and described in the Level I manual. The TRS-80 has two different sizes of characters which can be presented on the screen: 64-characters-per-line and 32-characters-per-line. It is this latter and the graphics which are used in the classroom.

c. cassette tape unit for program and data storage. The use of audio-cassette technology for program storage makes it convenient to transfer programs between teachers and schools. It also permits the development of inexpensive software on a commercial basis. The neophyte computer user, i.e. the classroom teacher, also enjoys having complete possession of his own program library. With audio-cassettes this is possible.

   Mini-diskette systems are becoming increasingly available also. The storage medium — the mini-diskette — is only slightly more expensive than the audio cassette. The advantage of the diskette system is the speed and reliability. However, they are also more expensive. To add a diskette unit to the TRS-80, an expansion interface must be added which has the port to which the diskette unit is attached. It is also recommended that 16K more of memory be added to the interface because the disk operating system uses a considerable portion of the 16K in the CPU.

d. the high-speed printer. The Quickprinter I, which is Radio Shack's version of the Centronics P-1, prints at 180 characters per second, prints both upper and lower case, and prints in three character sizes. Although this printer uses special paper, a sheet about 8½ inches long costs no more than a penny or two.

Besides the actual hardware, there are several other features of the TRS-80 which make it very attractive to the classroom teacher. These are:

a. portability. The TRS-80 with all the hardware mentioned above can be placed *in its entirety* on a cart and transported to different classrooms and schools. The only requirement to operate the system is an electrical outlet. Moving the computer to the students rather than vice versa is important since moving students from their regularly scheduled classrooms can be very disruptive.

b. ease of service. Any Radio Shack store in the country will accept a TRS-80 for service. They, in turn, will send it on to a local service center for repair — their responsibility, not yours. It may be possible to use a "loaner" while the other system is being serviced. This represents a tremendous convenience for teachers who are somewhat leery of the hardware in any case.

c. low-cost. A 32K TRS-80, Level II, plus monitor, cassette recorder, expander box, printer and mini-diskette unit costs under $3,000. *This is the cost for a complete system.* It is not necessary to have a service contract on the hardware because Radio Shack will repair any problems for a very nominal fee.

To estimate the "real" cost of this system to education, consider the following exercise. If we amortize the $3,000 over three years, assuming we use the hardware ten months per year and 20 days per month, we arrive at a figure of $5.00 per day for the hardware. If the computer is used 5 hours per day the cost is about $1.00 per hour for an entire class to use the personal computer. On a per student basis that averages out to about $.05 per student per hour.

STRATEGY 1 — THE "DYNAMIC BLACKBOARD"

The way a teacher uses a blackboard is the way a computer is used in the first strategy. Generally a blackboard is used to create a framework for classroom discussion — even if the "discussion" is primarily the teacher lecturing. Seeing the various salient points of the lecture in print on the blackboard helps the students understand.

Because setting up material on the blackboard can be time-consuming and tedious for the classroom teacher, alternatives to the blackboard have been developed. The most popular of these is the overhead projector. Its advantage over the blackboard is that materials for classroom discussion can be prepared ahead of time.

In many ways the use of the computer with a television monitor can be compared to the overhead projector in that the materials are prepared ahead of time. Also materials can either be prepared professionally or by the teacher himself. The computer has several advantages over the blackboards and overhead projectors. One advantage is convenience. When many overheads are to be used, there is much flipping and sorting of transparencies. "Flipping and sorting" with the computer are accomplished at the touch of a key. The computer programs are stored on audio-cassettes which take up far less room and are easier to carry than the bulky overhead transparencies. Furthermore the presentations on the computerized "blackboard" are dynamic. Animation and teacher-designed graphics bring unique situations to the "dynamic blackboard."

Because we live in a generation of students raised on television, the use of this medium seems to have an unexplainable hold on the students' attention. Students who would otherwise ignore the blackboard and even overhead transparencies will pay close attention to the television monitor. This is a phenomenon which crosses boundaries of subject matter as well as level of instruction.

Mathematics

In mathematics instruction we use the random number generator (the RND function) to generate problems which are presented on the television screen. The students are then asked to solve the problem in class. After a few moments (and under the teacher's control) the computer solves the problem step-by-step so that students can check if the way they solved the problem is correct. Because every step of the computer's output is under the control of the teacher, he is able to discuss each of the steps in solving the problem in as much detail as required. In other words, the computer is providing a framework for classroom discussion, just as the blackboard would.

But now, having completed one example the teacher can type RUN and receive another example which can be presented in as much detail (or less) as the first example. This process can be continued as long as the teacher feels that it is necessary. The teacher can bypass the explanations and use the problems generated for quizzes or for individualized work by students having difficulty.

Language Study

The "dynamic blackboard" can be used to teach foreign languages. For example, if Hebrew is taught, the graphics capability of the TRS-80 generates large Hebrew characters on the screen. The computer generates the conjugation of a particular root form and asks the students to identify the

tense and the person. Using animation the computer reinforces the correct answer by circling the various prefixes and suffixes that characterize this tense and person. A normally boring topic can be made fun using games, with two teams competing, and as an individual or small group tutorial. The computer is used to motivate drill-and-practice which is the essence of language study. The implications of this approach for study in any language including English are fairly obvious.

Science and Social Studies

In these areas the graphics capability of the TRS-80 to create diagrams and animate processes is used. For example, in social studies the TRS-80 is used to draw the map of a state (Ohio) and then quiz the students as to the location of various cities in the state. In general science, the TRS-80 may create a pump on the screen. The pump actually moves through its various cycles. As the basis for classroom discussion both of these applications of the "dynamic blackboard" have proven to be invaluable for motivating students and helping them learn.

We have only begun to describe the ways the dynamic blackboard can be used. The TRS-80-based dynamic blackboard has provided a motivational framework for instruction, in some cases it has provided a most effective learning experience, often it has provided an excellent reinforcing experience and a source of questions for quizzes in classroom evaluation.

STRATEGY 2 — "MATERIALS GENERATOR"

The classroom teacher spends considerable time preparing materials of one sort or another for distribution to the class. These materials can be homework assignments, a quiz or a test, or even something for students to work on for enrichment or remediation. The teacher may also prepare "fun"-type materials such as WORDFIND or CROSSWORD puzzles for use by the class. These serve to motivate students to undertake a reinforcing experience. To prepare such materials, the classroom teacher must first locate such materials, then transfer them to a medium appropriate for duplication. This process is of such importance to the classroom teacher that an entire extra-textbook materials industry has evolved selling such materials to teachers.

The TRS-80 with the Quickprinter can be used as a highly effective materials generator which is more cost-effective and more flexible than the materials that can be purchased in the teacher stores. While the idea of using the computer to generate classroom materials is not new, the "wrinkle" that is added to this strategy with the advent of the personal computer is convenience and accessability.

Mathematics

One of the ways in which we use the "materials generator" is to generate different problem sets for each student. The same time the computer generates the problem set for the student, it generates the solution for the teacher. One example is the output from such an application in the area of polynomial equations in intermediate algebra.

Sample of output #1

```
THE POLYNOMIAL FOR


SAM  SPERO

IS

 3           2
X  + -15.01 X  +  58.2358 X + -18.5867
```

```
THE POLYNOMIAL FOR

SAM  SPERO

IS

 3          2
X  + -15.01 X  +  58.2358 X + -18.5867

THE ROOTS ARE
 .35              6.54            8.12
THE EXTREMA ARE
   FOR X =   2.63238    Y =   48.9423
   FOR X =   7.37429    Y = -4.37031
```

The "materials generator" is also used to provide materials for several games such as MATHAGRAM and MATH-BINGO.

Language Study

The special types of materials which teachers prefer to prepare in language study lie primarily in the area of language games. The TRS-80 is especially strong in manipulating "strings." One of the popular games used extensively (and not only in language study) is WORDFIND puzzles. The teacher supplies a vocabulary list and within moments the computer generates a WORDFIND puzzle for use by the class.

Sample of output #2

```
&&&&1ST NUMBER IS THE ROW NUMBER
2ED  NUMBER IS THE COLUMN NUMBER
3RD NUMBER IS THE DIRECTION NUMBER
1 = DOWN   2 = RIGHT    3 = R-UP    4 = R-DOWN
5 = UP     6 = LEFT     7 = L-DOWN  8 = L-UP
  1 17 1   ALASKA          1 15 1   COLORADO
 20  8 2   DELAWARE       11  1 3   ILLINOIS
  6 12 4   INDIANA        10 11 4   KENTUCKY
  3  3 2   MONTANA        12  5 1   OHIO
 18 14 2   TEXAS           3 11 1   WYOMING

Z S P G D Y Q T F M R Z T X C S A Z Q   1
B B L N F W R L N J A P O U O I L O L   2
F Q M O N T A N A O W W Z M L F A N X   3
Z C X J N H Y S X I Y H Q T O W S Z Y   4
D T F X G Q I D H U O T K X R V K E X   5
V P W M M O V L E Z M I I X A L A X M   6
Q Z D N N S H D J N I E N I D Q M P O   7
W E V I M X K V G V N E S D O H P A X   8
Q V L K X G P U W W G W J C I Q X R C   9
N L W U J P C M C V K A Z K X A Y M S  10
I L K M F J R V V S M E M L I V N D X  11
I N K B O F K Y G U F F N T Y N D A T  12
X B F J H N Z Q F F X F Q T N Q T Z P  13
O I J L I Z P I T B H E N P U W B Q K  14
Y X W R O D Y Z W X D G N H G C Y R Y  15
S L C I F H I L D S J N O F F A K F M  16
A B Z R C Y L R T S S X P P P F Y Y Z  17
X Z R D D Q B W G F E U U T E X A S M  18
E J J U H P A V M T J C B P M X T J K  19
K W P Q I X V D E L A W A R E C G B P  20
  2   4   6   8  10  12  14  16  18  20
```

Another popular game approach used in language courses is the computerized MADLIB approach. We cannot show the entire dialogue that gives rise to the following, but we are able to generate uniquely tailored materials for each student using information supplied *by the student* as in Roger Price's original Madlibs.

Sample of output #3

```
HEERE IS THE STORY FOR

SAM  SPERO

ONCE UPON A TIME SAM SPERO AND KIT CARSON
CROSSED THE MISSISSIPPI RIVER ON THEIR WAY TO CALIFORNIA
SOON THEY SAW LARGE HERDS OF RATTLESNAKES
THEY SAW BLUE PAINTED INDIANS OF THE APACHE TRIBE
DARYLE WAS AFRAID AND RAN AWAY FROM OREGON
SAM STAYED AND WAS KILLED BY CHIEF CRAZY HORSE

CAN YOU FIND AT LEAST TWO ERRORS IN SPERO'S STORY?

PROVE THE ERRORS BY USE OF YOUR TEXTBOOK.
```

Science and Social Studies

In science and social studies, the computer is used to generate realistic data which can be analyzed by the students in order to apply procedures being studied, as well as to learn about various phenomena. For example, the famous Huntington II simulation packages have been adapted to use with the TRS-80. One of the simulations is the Millikan Oil Drop Experiment for determining the ratio of electric charge to the mass of the electron. The computer generates the actual data to be obtained from the experiment which the students can then analyze. The learning experience using the computer simulation is as good as that obtained in the laboratory experiment. By not having to do this experiment in the physics laboratory, the students have time to perform those experiments which lend themselves more to the actual lab experience.

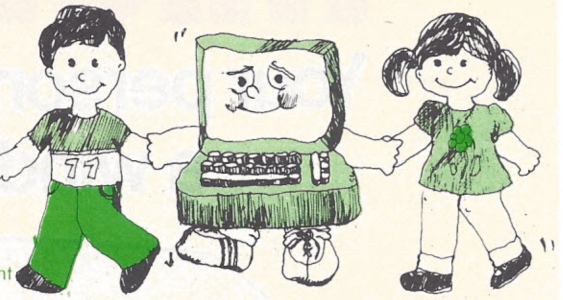
CONCLUSION

The two instructional strategies place the computer into an instructional role for which it is well-suited. In this role it provides the classroom teacher with technological assistance in completing tasks which the teacher cannot or will not perform himself. To provide this assistance, only a minimal computer facility is required — a so-called personal computer — which is relatively inexpensive. The teacher who uses the computer either as a "dynamic blackboard" or as a "materials generator" reaps maximum benefit.☐

My TRS-80 Likes Me

When I Teach Kids How to Use It

Part 10



By Bob Albrecht

WANDERING STAR, AGAIN

Last time we introduced you to Wandering Star, who wanders about the screen eating cosmic dust. Wandering Star first appears near the center of the screen, then rests for a brief time, perhaps thinking hungrily about cosmic dust.

After her brief rest, Wandering Star wanders. . .up, down, left, right. . .on the screen. If she should meander into a place that contains a cosmic dust mote, she eats it, and moves on.

Eventually, Wandering Star may reach the edge of the screen and disappear. This usually happens long before she has consumed all available cosmic dust.

If you tried our Wandering Star program, perhaps you notice that occasionally she wanders in a very strange way. Sometimes she jumps all the way from the left edge of the screen to the right edge, or from the right edge to the left edge. This happens, for example, if she is poised right at the edge of the screen at position 255 and tries to wander one place to the right to position 256. Position 256 is on the left edge of the screen, one line down from the line that includes position 255.

So we bring to you a new Wandering Star program, in which she wanders in a more mundane way without the benefit of hyperspace jumps.

```
100 REM***WANDERING STAR #2
110 CLS
200 REM***COSMIC DUST
210 FOR K = 1 TO 200
220 PRINT @ RND(1022), ".";
230 NEXT K
300 REM***WANDERING STAR APPEARS
310 ROW = 7
320 COL = 32
330 PRINT @(64*ROW + COL), "*";
400 REM***WANDERING STAR RESTS
410 T = 2000
420 FOR Z = 1 TO T : NEXT Z
500 REM***WANDERING STAR WANDERS
510 PRINT @(64*ROW + COL), " " ;        She leaves her old
520 W = RND(4)                                         place
530 IF W = 1 THEN ROW = ROW + 1   She might go down
540 IF W = 2 THEN ROW = ROW − 1   She might go up
550 IF W = 3 THEN COL = COL + 1   She might go right
560 IF W = 4 THEN COL = COL − 1   She might go left
600 REM***DID SHE GO OFF-SCREEN?
610 IF ROW < 0 OR ROW > 15 THEN 910
620 IF COL < 0 OR COL > 63 THEN 910
630 IF 64*ROW + COL = 1023 THEN 910
700 REM***NO, SHE IS STILL ON-SCREEN
710 PRINT @(64*ROW + COL), "*" ;
720 T = 100
730 FOR Z = 1 TO T : NEXT Z
740 GOTO 510
900 REM***WANDERING STAR GOES OFF-SCREEN
910 PRINT @ ), "WANDERING STAR HAS LEFT THIS
        UNIVERSE."
920 PRINT "FAREWELL, WANDERING STAR."
930 GOTO 930
```

Compare this program with our first Wandering Star program in the April 1980 issue of INTERFACE AGE. Last time, we thought of the screen as having 1024 print positions numbered from 0 to 1023. We avoided position 1023 because, if you print something there, everything on the screen scrolls up one line. Also see line 630 of our new program.

This time, we think of the screen as having 16 rows with 64 columns in each row. The rows are numbered from 0 (top row) to 15 (bottom row); the columns are numbered from 0 (left edge) to 63 (right edge).

16 rows × 64 columns = 1024 screen positions

In lines 330, 560 and 720, something is printed at a screen position in row ROW and column COL. For example,

```
330 PRINT @ 64*ROW + COL, "*";
```

In lines 510 through 560, Wandering Star wanders. First, she leaves the place where she was (line 510). Then she decides whether she will move down (W=1), or up (W=2), or right (W=3), or left (W=4).

She just might wander off-screen. This is checked by lines 610 and 620. If she wanders off-screen, the program jumps to line 910 and prints an appropriate message. This also happens if she wanders into position 1023 (line 630). Why? Because printing something in position 1023 causes everything on the screen to scroll up one line.

However, if she has stayed in the tiny universe of the TRS-80 screen, we must show her at her new place. This is done in lines 710 and 720. Finally, line 730 sends the computer back to let Wandering Star wander again.

THE RETURN OF WANDERING STAR

Well, Wandering Star wanders and. . .eventually. . . wanders off the screen, never again to appear on-screen. Farewell, Wandering Star.

Alas, the part of the universe surrounding the screen is a cosmic desert. The screen, of course, is a cosmic oasis.

So, after wandering in the desert for awhile, Wandering Star decides to return to the oasis where she can again savor cosmic dust and think about other oases elsewhere in the universe (and therein lies another story).

Think about how Wandering Star might return. She left the universe along one of the edges of the screen. . .Hmmm, perhaps she could reappear somewhere at the edge of the screen.

Does she learn from experience? Will she soon disappear again into the desert? Or will she remain in the food-rich oasis, pondering upon the greater universe, then invent or discover a way to move beyond the cosmic desert into other (and different) oases?

SELECTED SHORT SUBJECTS*

We call the following program "Countdown-Blastoff!" Enter it into your TRS-80 and RUN it.

```
100 REM***COUNTDOWN-BLASTOFF!
110 CLS
200 REM***COUNTDOWN FROM 10 TO 0
210 FOR C = 10 TO 0 STEP -1
220 PRINT C
230 FOR Z = 1 TO 300 : NEXT Z
240 NEXT C
250 PRINT "BLASTOFF!!!" : T = 400 : GOSUB 910
300 REM***SHOW SPACESHIP ON LAUNCH PAD
310 CLS
320 PRINT @512, "    *    "
330 PRINT       "   *U*   "
340 PRINT       "   *S*   "
350 PRINT       "   *A*   "
360 PRINT       "  *****  "
370 PRINT       "*******"
380 T = 400 : GOSUB 910
400 REM***LAUNCH THE SPACESHIP
410 PRINT       " !!! " : T = 300 : GOSUB 910
420 PRINT       " !!! " : T = 200 : GOSUB 910
430 PRINT       " !!! " : T = 100 : GOSUB 910
440 FOR K = 1 TO 16
450 PRINT : T = 100 : GOSUB 910
460 NEXT K
500 REM***ANNOUNCE A SUCCESSFUL LAUNCH
          AND STOP
510 CLS
520 PRINT "ALL SYSTEMS ARE GO. EVERYTHING IS
          AOK!"
530 END
900 REM***TIME DELAY SUBROUTINE
910 FOR Z = 1 TO T : NEXT Z
920 RETURN
```

Now that you are launched into space, play our simple reaction time game to pass time until you reach your destination.

```
100 REM***REACTION TIME PROGRAM
200 REM***INSTRUCTIONS TO THE PLAYER
210 CLS
220 PRINT "HOW FAST ARE YOU? I WILL CLEAR THE
230 PRINT "SCREEN FOR A LITTLE WHILE, THEN
240 PRINT "COUNT NEAR THE MIDDLE OF THE
250 PRINT "SCREEN. WHEN I START COUNTING,
260 PRINT "PRESS THE SPACE BAR AND I WILL
261 PRINT "STOP. STOP ME QUICKLY, IF YOU CAN!"
262 PRINT : PRINT "WHEN YOU ARE READY, PRESS
263 PRINT "ANY KEY"
270 IF INKEY$ = "" THEN 270
300 REM***CLEAR THE SCREEN FOR A RANDOM
          TIME, T
310 CLS
320 T = RND (2000)
330 FOR Z = 1 TO T : NEXT Z
400 REM***START COUNTING, SPACE BAR STOPS IT
410 X = 1
420 PRINT @472,X
430 IF INKEY<>" " THEN X = X + 1 : GOTO 420
500 REM***PLAYER PRESSED SPACE BAR. PAUSE,
          THEN PLAY AGAIN.
510 T = 2000
520 FOR Z = 1 TO T : NEXT Z
530 GOTO 210
```

*This section is excerpted from the book *TRS-80 BASIC: A Self-Teaching Guide* by Bob Albrecht, Don Inman and Ramon Zamora, copyright 1980 by John Wiley and Sons, Inc.

Play several times. An average of 10 is fast; congratulations. If your average is more than 20, well. . .maybe you are thinking about something else.

Humm. . .we played the game several times and discovered a way to cheat. We can stop the computer with a count of 1 every time. We can do this, *not* because we are that fast, but because there is a flaw in the program.

Beat the computer. Figure out how to stop the computer at 1 every time just by pressing the space bar. Later we will share our discovery with you, then show you how to fix the "bug" in the program so that this kind of cheating can't happen.

IMPORTANT. This computer error is not the fault of the computer. Rather, as are almost all computer errors, it is the fault of the programmer. This error almost escaped our notice. Imagine the letters we might have received if we had missed it.

COMPUTERTOWN, USA

This series is for teachers, parents or others who wish to help kids learn to use, program and enjoy computers. In ComputerTown, USA, kids do this in the public library.

We have found some especially good stuff to recommend to you.

Robert Purser's Magazine
P.O. Box 466, El Dorado, CA 95623
The most complete source of information about commercially available software for the TRS-80, Apple computers. Quarterly, $12/year.

CLOAD Magazine
Box 1267, Goleta, CA 93017
The best software bargain we know of for people who wish to help kids learn. This is a cassette magazine; each month you get a cassette with several programs to plug into your TRS-80. On a scale of one to ten dragonsmiles, CLOAD gets ten dragonsmiles. $36/year.

Adventure International
Box 3435, Longwood, FL 32750
The *Adventure* games from this company are one of the best ways we know for a child to learn and love problem-solving.

Cybernautics
Box 40132, San Francisco, CA 94140
Their games, TAIPAN and GALACTIC EMPIRE, are the best simulation games we have seen for a 16K TRS-80. Try 'em — you will never go back to Hammurabi, Kingdom, Star Trek, etc.

The Software Exchange
6 South St., Box 68, Milford, NH 03055
Perhaps the most complete source of cassette and disk software, books and magazines for your TRS-80. Ask for its catalog.

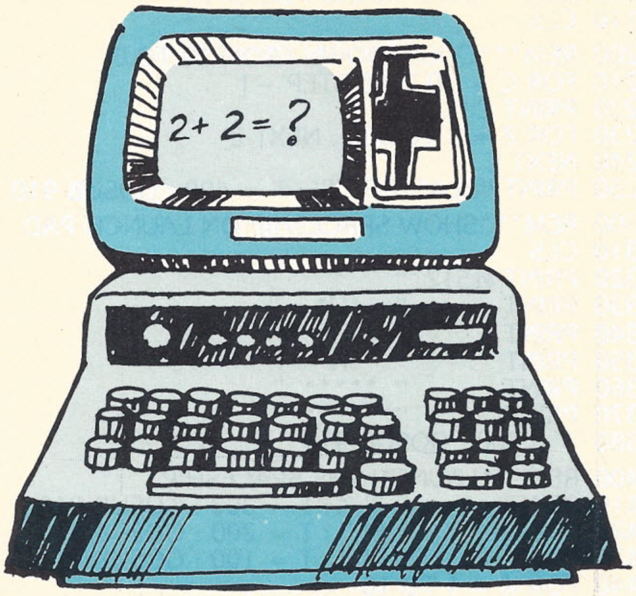
Computer Information Exchange
Box 158, San Luis Rey, CA 92068
CIE publishes a newspaper called *S-80 Bulletin, S-80 Computing* Magazine and *People's Software* — inexpensive cassettes with lots of programs.

And, as usual, for more info on ComputerTown, USA send a stamped, self-addressed envelope to ComputerTown, USA, P.O. Box 310, Menlo Park, CA 94025.□

Copies of "My TRS-80 Likes Me," Parts One and Two are now available free from Radio Shack. Interested persons should write to Sharon Ross, Radio Shack Circulation Dept. 3, 1300 One Tandy Center, Fort Worth, TX 76102.

Bob Albrecht can be contacted at P.O. Box 310, Menlo Park, CA 94025.

MATHTEST



By Bethany Prendergast

Microcomputers are particularly well adapted for use by the elementary school teacher. Their various uses in the role of C.A.I. (Computer Assisted Instruction) have been well documented by teachers and manufacturers alike. There is, however, another role for the microcomputer in the classroom that is passed over most probably because it lacks the glamor of the more exotic uses.

This role uses the microcomputer to relieve the teacher from the time-consuming administrative duties involved in the classroom. Nowhere is the need for this more evident than in the elementary math classes. The good teacher is faced with the monumental task of planning, writing, giving and correcting workpapers and tests.

This program is not just intended for teachers, however. It can very easily be used by parents who are concerned about their children's math. There are many times when parents want to drill their children in math fundamentals, but found either writing the tests or correcting them to be tedious work.

The need for the MATHTEST program came to me during this last year while I was introducing a course on microcomputing to the 7th and 8th graders at Assumption School in Jacksonville, Florida. The idea is to provide the teacher with a means to produce tests/workpapers almost at will. Each one can be made different, so there is no need to worry about cheating. The program is written for the TRS-80 Level II, but will operate on any 4K+ machine with very little modification. Just type it in and follow the prompt questions as they appear on the screen. There are remark lines throughout that explain what's happening.

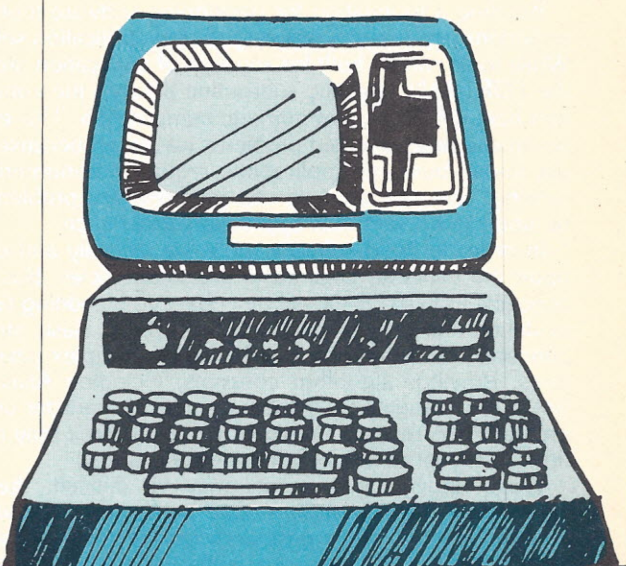
The program is designed to generate any number of multiplication, division, addition or subtraction test papers with up to 20 problems per page. It allows for 0-3 decimal places for the x and y, which are randomly selected numbers. The answers can either be placed on the same paper with the problems, or can be on a separate page. The answers that are generated will be to 4-place accuracy because of the print using statement. For more accuracy, that would have to be altered.

When the program is run, the user answers the prompt on the screen to determine the number of tests, the number of problems per page, the number of decimal places the x and y will have, and whether the test is to be multiplication, division, addition or subtraction.□

Program follows

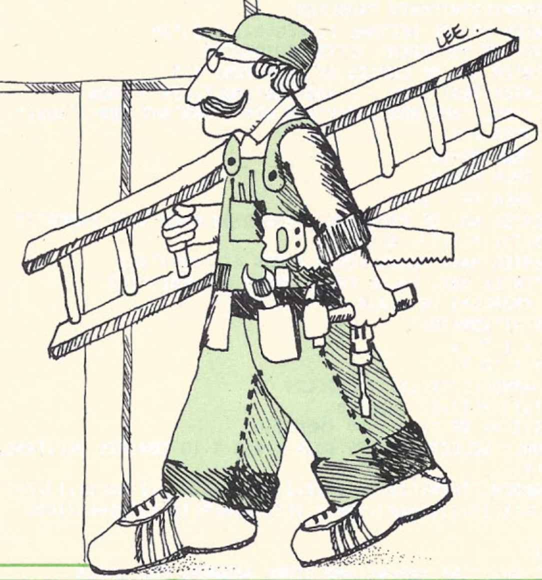
PROGRAM LISTING

```
10 '-PROGRAM LISTING FOR PUBLICATION - 08/01/79
20 '-PROGRAM NAME IS MATHTEST BY BETHANY PRENDERGAST
30 '-904/642-1902. WRITTEN FOR THE TRS-80 LEVEL II WITH LINE
40 '-PRINTER OF 80 COLUMNS. PROGRAM WILL GENERATE RANDOM TESTS
50 '-IN ACCORDANCE WITH ENTRIES FROM PROGRAM PROMPTS.
60 '-SET PP AHEAD OF P TO PREVENT DIV.BY 0 LATER
70 DEFDBL A, X, Y
80 CLS: RANDOM:PRINT"MATH PROBLEMS"
90 INPUT"ENTER NO. OF DECIMAL PLACES(0,1,2,3)";P
100 INPUT"NO. OF DIFFERENT TESTS DESIRED";S
110 INPUT"ENTER NO. OF COPIES OF EACH TEST";TD
120 INPUT"ENTER TEST TYPE - ( ADD,SUBT,MULT,DIV )";MD$
130 IF MD$<>"MULT"AND MD$<>"DIV"AND MD$<>"ADD"AND MD$<>"SUBT"  THEN 120
140 IF P=0 THEN PP=1
150 IF P=1 THEN PP=10
160 IF P=2 THEN PP=100
170 IF P=3 THEN PP = 1000
180 INPUT"ENTER NO. OF PROBLEMS DESIRED ON EACH TEST PAPER";T
190 DIM X(S,T), Y(S,T), A(S,T)
200 INPUT"ENTER MAX. VALUE FOR X (WHEN X/Y,X*Y)";L1
210 INPUT "ENTER MAX. VALUE FOR Y (WHEN X/Y,X*Y)";L2
220 'STORE PROBLEMS IN R.A.M.
230 CLS: PRINT"WORKING"
240 FOR II = 1 TO S
250 FOR I = 1 TO T
260 X(II,I)=RND(L1):Y(II,I)=RND(L2)
270 IF Y(II,I)>X(II,I) THEN260
280 IF X(II,I)=0 OR Y(II,I)=0 THEN 260
290 '-RANDOMLY SELECT X, Y OR BOTH X AND Y TO CONTAIN DEC.FRAC.
300 CH=RND(9)
310 IFCH>3ANDCH<7THENY(II,I)=Y(II,I)/PPELSEX(II,I)=X(II,I)/PP: Y(II,I)=Y(II,I)/PP
320 X(II,I)=(X(II,I)*1000)/1000: Y(II,I)=(Y(II,I)*1000)/1000
330 NEXT I
340 NEXT II
350 '-PRINT OUT TEST PAPERS AND STORE ANSWERS FOR LATER
360 IF MD$="MULT" THEN 400
370 IF MD$="DIV" THEN 580
380 IF MD$="ADD" THEN 980
390 IF MD$="SUBT" THEN 1180
400 FOR II = 1 TO S
410 FOR CC = 1 TO TD
420 LPRINT"                                          NAME:"
430 LPRINT"                                          DATE:"
440 LPRINT:LPRINT
450 LPRINT"                 MULTIPLICATION PROBLEMS   -     TEST PAPER # ";II
460 LPRINT:LPRINT
470 FOR I = 1 TO T
480 LPRINT"    ";I;")";X(II,I);"x";Y(II,I);"="
490 '-CALCULATE ANSWER AND STORE UNDER A(II,I)
500 A(II,I)=(X(II,I)*Y(II,I))
510 LPRINT
520 NEXT I
530 LS = 60 - ((T*2)+7)
540 FOR ZZ= 1 TO LS : LPRINT : NEXT ZZ
550 FOR ZZ= 1 TO 2000:NEXT ZZ
560 NEXT CC:NEXT II
570 GOTO 750
580 FOR II = 1 TO S
590 FOR  CC= 1 TO TD
600 LPRINT"                                          NAME:"
610 LPRINT"                                          DATE:"
620 LPRINT:LPRINT
630 LPRINT"                 DIVISION PROBLEMS  -  TEST PAPER # ";II
640 LPRINT:LPRINT
650 FOR I= 1 TO T
660 LPRINT"    ";I;")";X(II,I);"/";Y(II,I);"="
670 '-COMPUTE ANSWER AND STORE UNDER A(II,I)
680 A(II,I)=(X(II,I)/Y(II,I))
690 LPRINT
700 NEXT I
710 LS = 60 - ((T*2)+7)
720 FOR ZZ = 1 TO LS : LPRINT : NEXT ZZ
730 FOR ZZ = 1 TO 2000 : NEXT ZZ
740 NEXT CC:NEXT II
750 IF MD$="MULT" THEN LPRINT"   ANSWERS TO MULTIPLICATION TESTS"
760 IF MD$="DIV" THEN LPRINT"   ANSWERS TO DIVISION TESTS"
770 IF MD$="ADD" THEN LPRINT"   ANSWERS TO ADDITION TESTS"
780 IF MD$="SUBT" THEN LPRINT"   ANSWERS TO SUBTRACTION TESTS"
790 LPRINT
800 '-A$ AND C$ ARE USED TO KEEP ANSWERS IN COLUMNS
810 A$="      "
820 C$="     "
830 B$="#,###,###.####"
840 FOR II= 1 TO S: LPRINT"          TEST PAPER # ";II
850 LPRINT
860 '-KEEP CHECKING THAT THERE IS AN A TO PRINT
870 FOR I= 1 TO T  STEP 3
880 IF I<10 THEN X$=A$ ELSE X$=C$
890 LPRINT X$;I;")";USINGB$;A(II,I);
900 IF I = T THEN 960
910 IF I+1<10 THEN X$=A$ ELSE X$=C$
920 LPRINT X$;I+1;")";USINGB$;A(II,(I+1));
930 IF I+1 =T THEN 960
940 LPRINT X$;I+2;")";USINGB$;A(II,(I+2))
950 IF I+2<10 THEN X$=A$ ELSE X$=C$
960 NEXT I:LPRINT:LPRINT:NEXT II
970 END
980 FOR II=1 TO S
990 FOR CC=1 TO TD
1000 LPRINT TAB(45)"NAME:"
1010 LPRINT TAB(45)"DATE:"
1020 LPRINT:LPRINT
1030 LPRINTTAB(20)"ADDITION PROBLEMS-TEST PAPER #";II
1040 LPRINT:LPRINT
1050 FOR I=1 TO T
1060 LPRINT"    ";I;")"X(II,I);"+";Y(II,I);"="
1070 'COMPUTE ANSWER AND STORE UNDER A(II,I)
1080 A(II,I)=X(II,I)+Y(II,I)
1090 LPRINT
1100 NEXT I
1110 ' GENERATE LINE FEEDS FOR NEXT FORM
1120 LS=60-((T*2)+7)
1130 FOR ZZ=1 TO LS:LPRINT:NEXT ZZ
1140 'GENERATE FALSE LOOP BETWEEN PRINTS
1150 FOR ZZ=1 TO 2000:NEXTZZ
1160 NEXT CC: NEXT II
1170 GOTO750
1180 FOR II=1 TO S
1190 FOR CC=1 TO TD
1200 LPRINT TAB(45)"NAME:"
1210 LPRINT TAB(45)"DATE:"
1220 LPRINT:LPRINT
1230 LPRINTTAB(20)"SUBTRACTION PROBLEMS-TEST PAPER #";II
1240 LPRINT:LPRINT
1250 FOR I=1 TO T
1260 LPRINT"    ";I;")"X(II,I);"-";Y(II,I);"="
1270 'COMPUTE ANSWER AND STORE UNDER A(II,I)
1280 A(II,I)=X(II,I)-Y(II,I)
1290 LPRINT
1300 NEXT I
1310 ' GENERATE LINE FEEDS FOR NEXT FORM
1320 LS=60-((T*2)+7)
1330 FOR ZZ=1 TO LS:LPRINT:NEXT ZZ
1340 'GENERATE FALSE LOOP BETWEEN PRINTS
1350 FOR ZZ=1 TO 2000:NEXTZZ
1360 NEXT CC: NEXT II
1370 GOTO750
```



Using and Building Micro-Based Systems

Chapter Six



By David Marca, Associate Editor

This chapter develops the concept of software tools and provides some useful examples in FORTRAN. The reader will find an orderly progression from simple concepts to complex examples. The organization of the article allows for the reading and programming of independent tools that are used in a building-block manner (one on top of the other) to create useful applications.

FOUNDATIONS FOR SOFTWARE TOOLS

Building a foundation for developing software tools is an important step towards well-engineered application software. While tools can be built for any type of application area (like the FORTRAN scientific subroutine library), the concentration here will be on non-numeric computation. This emphasis on character-oriented problems was taken because many day-to-day business applications require non-numeric processing, and also because character-oriented problems can be understood with little application experience.

In order to build software tools, we will rely and expand upon the concepts covered in previous articles. Basic data concepts in FORTRAN will be enhanced by adding facilities to define and manipulate character strings. Basic structure concepts will be utilized to build more complex data structures. Previous algorithm concepts, including Alias/Alibi, will be reinvestigated in the new context of character-oriented problems. (A review of chapter five at this time may help to firm up fundamentals of computer programs.)

While underlying concepts may be difficult, the tools approach is rather simple — one builds upon the work of others. Not everyone's past work can be reused, however. Traditionally, a program or a subroutine is constructed to perform one or (at best) a few specific jobs. A software tool, however, must be able to work in a multitude of different environments. It should have well-defined, simple, and standard interfaces, and avoid the idiosyncrasies of FORTRAN, concealing the evident ones in select modules.

Lastly, but most importantly, the program or subroutine should perform a very specific (at best, a single) function on a small set of data. Ideally, this type of software has a well-chosen set of filters that becomes a set of building blocks which can be used separately or together, handling many programming needs.

These blocks improve the quality of application software in a micro environment. Smaller programs and subroutines decrease the potential for errors in testing and during actual use. Also, limitations in programming languages and hardware can be hidden from most of the application by the use of filters. For example, the isolation of all I/O details in a set of subroutines requires changes to only those subroutines when the software is moved to a different hardware configuration.

Good tools are also important when creating a micro-based software system. Current micro-environments are sorely lacking in adequate tools to produce well-engineered software. Four such programming aids are:

- Trace facility
- File comparator
- Call and called by maps
- Documentation extraction from source code

After our software tool foundations have been established, the above types of programming aids should be easy to implement on most microcomputers having a FORTRAN compiler. Since there is a heavy emphasis on non-numeric processing, we must now investigate how the FORTRAN language handles character strings.

FORTRAN CHARACTER STRINGS

These capabilities were added to the original FORTRAN II language in March 1966 when the FORTRAN IV standards were approved. Hollerith constants (alphanumeric characters surrounded by single quotes) were allowed to be used in very limited contexts. Each numeric data type was allowed to contain character information. The maximum number of characters held by a data type depended upon the hardware's byte length and storage allocation rules of the compiler. A rule of thumb is one character per byte size of the data type (figure 1).

VARIABLE TYPE	SIZE (BYTES)	MAXIMUM CHARACTERS ALLOWED
DOUBLE PRECISION	8	8
REAL	4	4
INTEGER	2	2
LOGICAL	1	1

Figure 1. Character capacity of Fortran data types (for TRS-80).

Within a data type, characters can be packed using different formats. The programmer is allowed to read in a write-out character using the "A" format code. One to eight characters can be transferred to or from a variable by specifying A1 to A8 respectively in a FORMAT statement (figure 2). During input from a device, the characters are right justified in the variable (right-most character is put in the right-most byte, etc.).

```
      INTEGER TTY,KEY
      LOGICAL CHAR1
      INTEGER CHAR2
      REAL CHAR3,CHAR4
      DOUBLE PRECISION CHAR5,CHAR6,CHAR7,CHAR8
      DATA TTY/1/,KEY/1/
      WRITE(TTY,50)
50    FORMAT(1H ,'START OF PROGRAM')
      READ(KEY,100) CHAR1
100   FORMAT(1A1)
      READ(KEY,200) CHAR2
200   FORMAT(1A2)
      READ(KEY,300) CHAR3
300   FORMAT(1A3)
      READ(KEY,400) CHAR4
400   FORMAT(1A4)
      READ(KEY,500) CHAR5
500   FORMAT(1A5)
      READ(KEY,600) CHAR6
600   FORMAT(1A6)
      READ(KEY,700) CHAR7
700   FORMAT(1A7)
      READ(KEY,800) CHAR8
800   FORMAT(1A8)
      WRITE(TTY,9999)
9999  FORMAT(1H ,'END OF PROGRAM')
      STOP
      END
```

Figure 2. Reading in characters.

Characters are also allowed to be put into a variable by the DATA statement. The programmer can specify a Hollerith constant with a size equal to or less than the character capacity of the variable. On some compilers, the DATA statement left-justifies characters in the variable. This has been a constant source of programmer errors. Try out the program in figure 3 to see if the DATA statement right or left justifies on your compiler.

```
      INTEGER TTY,KEY
      INTEGER INIT,INPUT
      DATA TTY/1/,KEY/1/
      DATA INIT/'A'/
      WRITE(TTY,100)
100   FORMAT(1H ,'START OF PROGRAM')
      WRITE(TTY,200)
200   FORMAT(1H ,'PLEASE ENTER THE LETTER A - ')
      READ(KEY,300) INPUT
300   FORMAT(1A1)
      WRITE(TTY,400) INIT,INPUT
400   FORMAT(1H ,'INIT=',1A1,' INPUT=',1A1)
      IF (.NOT.(INPUT .EQ. INIT)) GOTO 600
         WRITE(TTY,500)
500      FORMAT(1H ,'DATA STATEMENT RIGHT-JUSTIFIES')
         GOTO 800
600   CONTINUE
         WRITE(TTY,700)
700      FORMAT(1H ,'DATA STATEMENT LEFT-JUSTIFIES')
800   CONTINUE
      WRITE(TTY,9999)
9999  FORMAT(1H ,'END OF PROGRAM')
      STOP
      END
```

Figure 3. Packing of characters—data statement vs. read statement.

```
      INTEGER TTY,KEY,PTR
      INTEGER BLANK,CHAR
      DATA TTY/1/,KEY/1/,PTR/2/,BLANK/8224/
C
      WRITE(TTY,100)
100   FORMAT(1H ,'START OF PROGRAM')
      WRITE(PTR,200)
200   FORMAT(1H ,'---------------------')
      WRITE(PTR,300)
300   FORMAT(1H ,'CHARACTER   CODE(1A1)')
      WRITE(PTR,200)
C
      WRITE(TTY,400)
400   FORMAT(1H ,'?')
      READ(KEY,500) CHAR
500   FORMAT(1A1)
C
600   IF (CHAR .EQ. BLANK) GOTO 800
         WRITE(PTR,700)CHAR,CHAR
700      FORMAT(1H ,4X,1A1,8X,I6)
         WRITE(TTY,400)
         READ(KEY,500) CHAR
         GOTO 600
800   CONTINUE
      WRITE(PTR,200)
C
      WRITE(TTY,9999)
9999  FORMAT(1H ,'END OF PROGRAM')
      STOP
      END
```

Figure 4. DECODE Program.

The last way characters can be stored into a variable is by using a Hollerith constant as an argument in a subprogram call. According to subprogram calling conventions, a variable within a subprogram acquires those characters as if they had been read in or initialized with a DATA statement. The variable's content can then be written out, compared, or manipulated as desired.

The actual contents of any variable containing characters can be displayed in numeric form as well as character form. Characters can be read in and written out using the "A" format, or their numeric equivalent can be stored or retrieved using any FORTRAN numeric operation. A quick way to document the numeric equivalent for each character on your machine is to run the DECODE program (figure 4). This program reads in characters in "A1" format, and prints the characters and their numeric equivalent on the printer. Once numeric equivalents are known, character sets (e.g., upper case, lower case, special) can be defined numerically for an application (this program was used to specify character sets for the string package presented further in this article).

```
MAIN PROGRAM
   CALL SETTRC (trace parameters)
   CALL A
   CALL B
   STOP
   END
```

```
SUBROUTINE A
   CALL CALL ('A')
   .
   .
   .
   CALL RETR ('A')
   RETURN
   END
```

```
SUBROUTINE B
   CALL CALL ('B')
   .
   .
   CALL C
   .
   .
   CALL RETR ('B')
   RETURN
   END
```

```
SUBROUTINE C
   CALL CALL ('C')
   .
   .
   .
   .
   CALL RETR ('C')
   RETURN
   END
```

Figure 5. Using the trace facility.

TRACE FACILITY

Even with the limited character string capabilities of FORTRAN, we can build one of the four important tools mentioned earlier — the trace facility. This records each subprogram's CALL and RETURN. The trace facility built for this tutorial was designed to be invoked immediately before and after any subprogram code is executed. Thus, one part of the tool must record the fact that a particular subprogram was called, and another part must signal that the subprogram is returning to its caller. The subroutines "CALL" and "RETR" provide these functions respectively.

In order to use the trace facility, a subprogram must invoke the "CALL" subroutine before it starts to do any work. The logic of a subprogram should be structured so that there is always only one entry point and one exit point to the subprogram. In this way, the subprogram need only call the "CALL" subroutine at the beginning, and the "RETR" subroutine at the end. The argument to both "CALL" and "RETR" is a single Hollerith constant representing the subprogram name. The mainline program is required to initialize the tool by specifying the trace device and a trace on/off switch (figure 5).

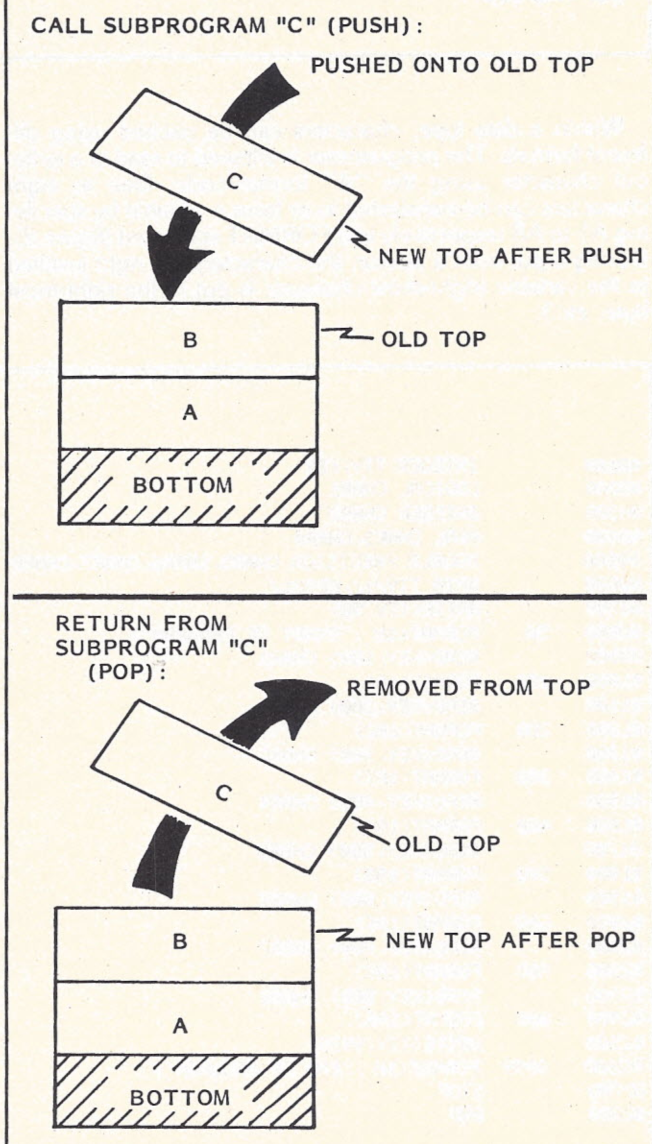


Figure 6. Trace stack data structure and operations.

The overall design and implementation of the trace tool was very dependent upon the nature of the problem. Keeping track of calls and returns requires the saving of the current subprogram name when another (child) call is made, and the restoration of the caller (parent) subprogram name upon a return. A very common data structure which can handle this type of problem is called a stack. The term "stack" is given because the information in the data structure is manipulated like a stack of coins. Only the top of the stack can be touched — either to add a new piece of data (push) or remove the piece of data at the top (pop). Thus a call to a subprogram would push that subprogram's name on the top of the stack, and a return would require the removal of that name from the stack top (figure 6). Having a data structure and its operations so closely resembling the problem structure made the design and coding (figure 7) easy. A simple one-dimensional array (stack) keeps the subprogram names, while a single scalar variable (level) keeps track of the top of the stack (i.e., last assigned element in the array).

```
      SUBROUTINE CALL(NAME)
      DOUBLE PRECISION NAME
      LOGICAL TRACE
      INTEGER LEVEL,STKSIZ,DEVICE
      DOUBLE PRECISION STACK(20)
      COMMON /TRCSTK/TRACE,DEVICE,LEVEL,STKSIZ,STACK
      LEVEL=LEVEL+1
      IF (.NOT.(LEVEL.LE.STKSIZ)) GOTO 300
         STACK(LEVEL)=NAME
         IF (.NOT.(TRACE)) GOTO 200
            WRITE(DEVICE,100) LEVEL,NAME
100         FORMAT(1H ,I3,' CALL: ',1A8)
200      CONTINUE
300   CONTINUE
      RETURN
      END
C
      SUBROUTINE RETR(NAME)
      DOUBLE PRECISION NAME
      LOGICAL TRACE
      INTEGER LEVEL,STKSIZ,DEVICE
      DOUBLE PRECISION STACK(20)
      COMMON /TRCSTK/TRACE,DEVICE,LEVEL,STKSIZ,STACK
      IF (.NOT.(LEVEL.GT.0)) GOTO 700
         IF (.NOT.(LEVEL.LE.STKSIZ)) GOTO 500
            IF (.NOT.(STACK(LEVEL).EQ.NAME)) GOTO 200
               IF (.NOT.(TRACE)) GOTO 150
                  WRITE(DEVICE,100) LEVEL,NAME
100               FORMAT(1H ,I3,' RETR: ',1A8)
150            CONTINUE
                  LEVEL=LEVEL-1
                  GOTO 400
200         CONTINUE
               WRITE(DEVICE,300) NAME,STACK(LEVEL)
300            FORMAT(1H ,'RETR: ',1A8,' INSTEAD: ',1A8)
               LEVEL=LEVEL-1
400         CONTINUE
            GOTO 600
500      CONTINUE
            LEVEL=LEVEL-1
600      CONTINUE
         GOTO 900
700   CONTINUE
         WRITE(DEVICE,800) NAME
800      FORMAT(1H ,'NO MATCHING CALL FOR: ',1A8)
900   CONTINUE
      RETURN
      END
C
      SUBROUTINE SETTRC(TRCON,TRCDEV)
      LOGICAL TRCON
      INTEGER TRCDEV
      LOGICAL TRACE
      INTEGER LEVEL,STKSIZ,DEVICE
      DOUBLE PRECISION STACK(20)
      COMMON /TRCSTK/TRACE,DEVICE,LEVEL,STKSIZ,STACK
      TRACE=TRCON
      DEVICE=TRCDEV
      LEVEL=0
      STKSIZ=20
      RETURN
      END
```

Figure 7. Trace facility code.

CHARACTER STRING HANDLER

Clearly, FORTRAN is not one of the richest languages when it comes to character string handling. The current standard does not even have character data types or operations. Better string processing is, however, being considered for the next version of the industry standard. Even with present deficiencies, some have proposed techniques for represent-

FUNCTION	CTRAN	BASIC (level II)
Initialize	CALL INISTR	not needed
Declare string	CALL DCLSTR(A,50)	not needed
Read string	CALL GETSTR(TTY,A)	INPUT A$
Write string	CALL PUTSTR(TTY,A)	PRINT A$
Assign constant	no equivalent	A$="LITERAL CONSTANT"
Assign string	CALL MOVSTR(B,A)	B$=A$
Concatenate	CALL MOVSTR(B,CATSTR(A,C))	B$=A$+C$
Repeat	CALL MOVSTR(B,REPSTR(A,5))	partial functionality with STRING$
Take a Piece	CALL MOVSTR(B,SUBSTR(A,3,7))	B$=MID$(A$,3,7)
Change a Piece	CALL MOVSTR(SUBSTR(A,2,5),B)	no equivalent
Length of String	S=LENSTR(A)	S=LEN(A$)
Find start of a pattern	P=FNDSTR(A,B)	no equivalent
Null string	CALL NULSTR(A)	A$=""
Blank string	CALL BLKSTR(A)	no equivalent
Get special character set	CALL GETSPL(A)	A$="!@#$%∧&*()+_:.,;?/\'<>"
Compare for equality	IF(EQSTR(A,B))...	IF A$=B$...
Compare for less than	IF(LTSTR(A,B))...	IF A$<B$...
Compare for greater than	IF(GTSTR(A,B))...	IF A$>B$...
Set to HIGH VALUES	CALL GETHGH(A)	no equivalent
Set to LOW VALUES	CALL GETLOW(A)	

Figure 8. CTRAN interfaces compared with BASIC.

ing variable length character strings. Others have expanded upon these earlier concepts and developed a good set of character handling subroutines.

There are many good reasons for developing this type of software tool, especially in the micro-environment. A string package generalizes the local environment around an application program, providing a simple and consistent level of interface. In this way, the application logic is only concerned with the manipulation of character strings as it relates to the problem solution. Details about the form of storage and I/O are hidden from the application level. Certainly these qualities enhance the ability to move programs from one system or compiler to another with little or no change — a distinct advantage of FORTRAN.

A good character string package for FORTRAN must provide several functions:

- declare a string variable
- read in/write out a string from/to a device
- build a string from other strings
- break a string apart
- scan a string for a pattern
- determine a string's length
- compare strings to each other

The character string package developed for this tutorial provides all the above functions, plus some important utilities. Interfaces shown in figure 8 were forced to be implemented by subroutines, however, due to compiler limitations. This package will be called CTRAN in this series.

A simple test showing how an application program actually uses the CTRAN package is in figure 9. Here, simple read-first logic is used to read in a string and then echo what was entered. The program stops when no characters are entered. Tracing is initialized since all tools in this tutorial use that facility to help the programmer test an application.

```
      INTEGER TTY,KEY
      INTEGER STRING
      LOGICAL TRACE
      DATA TTY/1/,KEY/1/
C
      WRITE(TTY,100)
100   FORMAT(1H ,'START OF PROGRAM')
      TRACE=.TRUE.
      CALL SETTRC(TRACE,TTY)
      CALL INISTR
      CALL DCLSTR(STRING,30)
C
      WRITE(TTY,200)
200   FORMAT(1H ,'STRING?')
      CALL GETSTR(KEY,STRING)
C
300   IF (0 .EQ. LENSTR(STRING)) GOTO 500
        WRITE(TTY;400)
400     FORMAT(1H ,'STRING ENTERED...')
        CALL PUTSTR(TTY,STRING)
        WRITE(KEY,200)
        CALL GETSTR(KEY,STRING)
        GOTO 300
500   CONTINUE
C
      WRITE(TTY,9999)
9999  FORMAT(1H ,'END OF PROGRAM')
      STOP
      END
```

Figure 9. Echo Program.

Experience with building and using a character handling package of this type has identified three important optimizations for micros. First, any portion dealing with input/output can be recoded to take advantage of a particular machine, compiler and storage size. This will increase the speed of the software if I/O routines have very general algorithms. Second, CTRAN uses "A1" format with an INTEGER array for its central string space. Recoding to use the most storage efficient data type for a particular machine could certainly reduce wasted storage. Lastly, all internal operations should be done with the fastest INTEGER arithmetic possible. This may be in conflict with the previous recommendation, and usually space is more of a problem than small inefficiencies in speed.

WORD ISOLATOR

Now we have enough tools to make our first set of real filters. The string handling package divorces us from the details of variable length character strings, and allows us to concentrate on application details. The trace facility will be employed to help test the software we build. Keeping in mind those four major programming aids mentioned in the introduction, we can build a level of filters which provide a common level of support — a word isolator.

We can define a "word" as being a collection of alphanumeric characters surrounded by special characters. Special characters are symbols not commonly found in a word (e.g., comma, period, colon, question mark, blank, etc.). The actual text containing those words should be considered as a single stream of characters to the word isolator. This allows the details of how the text is actually stored to be hidden from the real problem of forming a word. We therefore have two major parts to the problem. The first is forming words from a stream of characters. The second involves creating a stream of characters from some sort of physical storing of text.

Having stated our two major requirements, we can design these two important parts (figures 10 and 11). The design representation used is the Nassi-Schneiderman Structured Flowchart forms. For this exercise, we are not taking into account the problem of detecting when the entire input is exhausted. We are also limiting our input device to the keyboard, both for simplicity and to allow you to easily implement these programs.

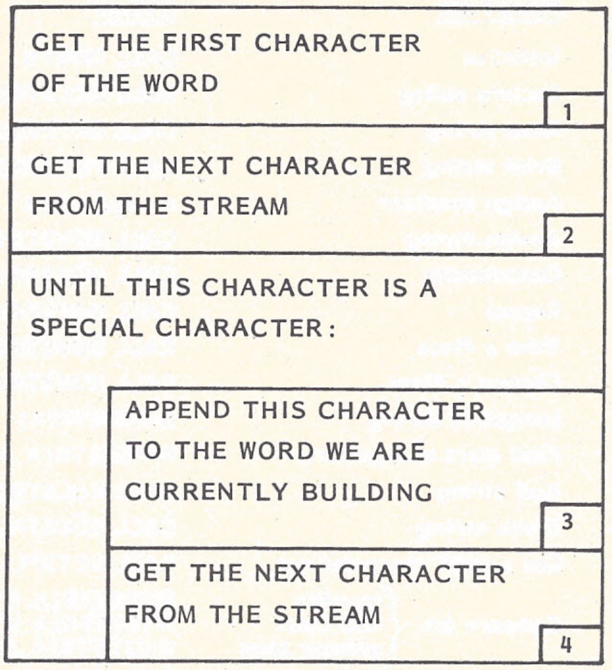


Figure 10. Design of Get Word filter.

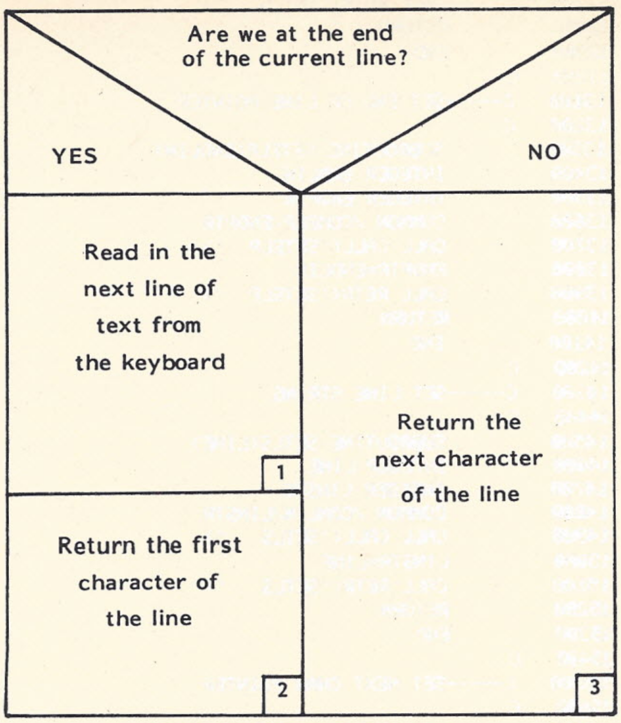


Figure 11. Design of Get Next Character filter.

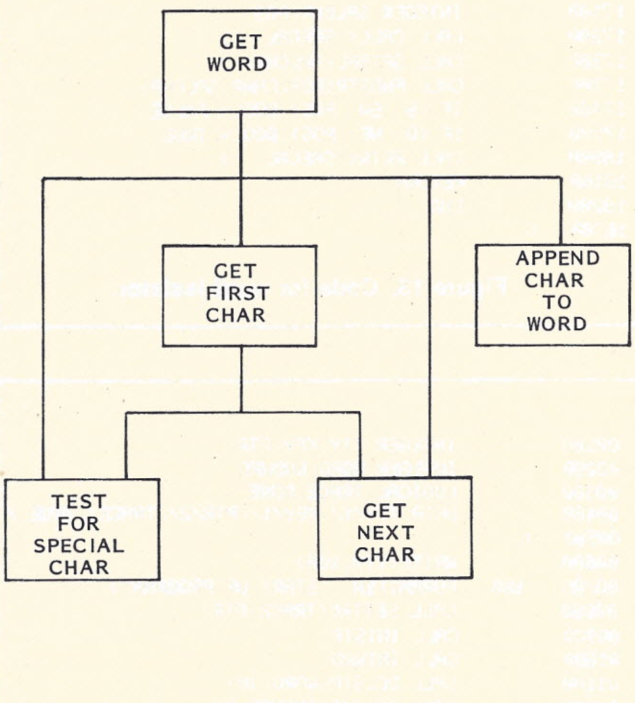


Figure 12. Word isolator general design.

Using these two filters, a general design was established (figure 12). This design representation uses Hierarchical Input Process Output diagrams, including all the major functions required to perform word isolation. During implementation, this design was used to develop the major subprograms. Other subprograms were built to provide a clean way of remembering and retrieving important information shared among many subprograms (figure 13). Additional design was needed to define these, but discussion of this is beyond the scope of this tutorial. A simple test of the word isolator shows how easy it is to get a word from text with this filtering mechanism (figure 14).

```
C-----GET CHAR STRING VARIABLE
C
      SUBROUTINE GETCS(CHAR)
      INTEGER CHAR
      INTEGER CHRSTR
      COMMON /COMCS/CHRSTR
      CALL CALL('GETCS   ')
      CHAR=CHRSTR
      CALL RETR('GETCS   ')
      RETURN
      END
C
C-----GET END OF LINE POINTER
C
      SUBROUTINE GETELP(ENDLIN)
      INTEGER ENDLIN
      INTEGER ENDPTR
      COMMON /COMELP/ENDPTR
      CALL CALL('GETELP  ')
      ENDLIN=ENDPTR
      CALL RETR('GETELP  ')
      RETURN
      END
C
C-----GET FIRST CHAR IN WORD
C
      SUBROUTINE GETFCR(DEVICE,CHAR)
      INTEGER DEVICE,CHAR
      LOGICAL SPLCHR
      CALL CALL('GETFCR  ')
      CALL GETNCR(DEVICE,CHAR)
      CALL SPECAL(SPLCHR,CHAR)
100   IF (.NOT.SPLCHR) GOTO 200
         CALL GETNCR(DEVICE,CHAR)
         CALL SPECAL(SPLCHR,CHAR)
         GOTO 100
200   CONTINUE
      CALL RETR('GETFCR  ')
      RETURN
      END
C
C-----GET LINE STRING
C
      SUBROUTINE GETLS(LINE)
      INTEGER LINE
      INTEGER LINSTR
      COMMON /COMLIN/LINSTR
      CALL CALL('GETLS   ')
      LINE=LINSTR
      CALL RETR('GETLS   ')
      RETURN
      END
C
C-----GET NEXT CHAR POINTER
C
      SUBROUTINE GETNCP(CHRPTR)
      INTEGER CHRPTR
```

```
      INTEGER NXTPTR
      COMMON /COMNCP/NXTPTR
      CALL CALL('GETNCP  ')
      CHRPTR=NXTPTR
      CALL RETR('GETNCP  ')
      RETURN
      END
C
C-----GET NEXT CHAR
C
      SUBROUTINE GETNCR(DEVICE,CHAR)
      INTEGER DEVICE,CHAR
      INTEGER LINE,PIECE
      INTEGER CURCHR,ENDLIN
      CALL CALL('GETNCR  ')
      CALL GETLS(LINE)
      CALL GETNCP(CURCHR)
      CALL GETELP(ENDLIN)
      IF (.NOT.(CURCHR.GE.ENDLIN)) GOTO 100
         CALL GETSTR(DEVICE,LINE)
         CURCHR=0
100   CONTINUE
      CURCHR=CURCHR+1
      CALL SETNCP(CURCHR)
      CALL SUBSTR(PIECE,LINE,CURCHR,1)
      CALL MOVSTR(CHAR,PIECE)
      CALL RETR('GETNCR  ')
      RETURN
      END
C
C-----GET WORD
C
      SUBROUTINE GETWRD(DEVICE,WORD)
      INTEGER DEVICE,WORD
      INTEGER CHAR,MORE
      LOGICAL SPLCHR
      CALL CALL('GETWRD  ')
      CALL GETCS(CHAR)
      CALL GETFCR(DEVICE,CHAR)
      CALL MOVSTR(WORD,CHAR)
      CALL GETNCR(DEVICE,CHAR)
      CALL SPECAL(SPLCHR,CHAR)
100   IF (SPLCHR) GOTO 200
         CALL CONCAT(MORE,WORD,CHAR)
         CALL MOVSTR(WORD,MORE)
         CALL GETNCR(DEVICE,CHAR)
         CALL SPECAL(SPLCHR,CHAR)
         GOTO 100
200   CONTINUE
      CALL RETR('GETWRD  ')
      RETURN
      END
C
C-----INIT GET WORD PACKAGE
C
      SUBROUTINE INIWRD
      INTEGER CHAR,LINE
      INTEGER LINSIZ
      CALL CALL('INIWRD  ')
      CALL DCLSTR(CHAR,1)
      CALL SETCS(CHAR)
      LINSIZ=80
      CALL DCLSTR(LINE,LINSIZ)
      CALL SETLS(LINE)
      CALL SETELP(LINSIZ)
      CALL SETNCP(LINSIZ+1)
      CALL RETR('INIWRD  ')
      RETURN
      END
C
C-----SET CHAR STRING VARIABLE
C
      SUBROUTINE SETCS(CHAR)
      INTEGER CHAR
      INTEGER CHRSTR
      COMMON /COMCS/CHRSTR
      CALL CALL('SETCS   ')
      CHRSTR=CHAR
      CALL RETR('SETCS   ')
      RETURN
      END
C
C-----SET END OF LINE POINTER
C
      SUBROUTINE SETELP(ENDLIN)
      INTEGER ENDLIN
      INTEGER ENDPTR
      COMMON /COMELP/ENDPTR
      CALL CALL('SETELP  ')
      ENDPTR=ENDLIN
      CALL RETR('SETELP  ')
      RETURN
      END
C
C-----SET LINE STRING
C
      SUBROUTINE SETLS(LINE)
      INTEGER LINE
      INTEGER LINSTR
      COMMON /COMLIN/LINSTR
      CALL CALL('SETLS   ')
      LINSTR=LINE
      CALL RETR('SETLS   ')
      RETURN
      END
C
C-----SET NEXT CHAR POINTER
C
      SUBROUTINE SETNCP(CHRPTR)
      INTEGER CHRPTR
      INTEGER NXTPTR
      COMMON /COMNCP/NXTPTR
      CALL CALL('SETNCP  ')
      NXTPTR=CHRPTR
      CALL RETR('SETNCP  ')
      RETURN
      END
C
C-----TEST FOR SPECIAL CHAR
C
      SUBROUTINE SPECAL(BOOL,CHAR)
      INTEGER CHAR
      LOGICAL BOOL
      INTEGER SPLCHR,POS
      CALL CALL('SPECAL  ')
      CALL GETSPL(SPLCHR)
      CALL FNDSTR(POS,CHAR,SPLCHR)
      IF (0.EQ.POS) BOOL=.FALSE.
      IF (0.NE.POS) BOOL=.TRUE.
      CALL RETR('SPECAL  ')
      RETURN
      END
C
```

Figure 13. Code for word isolator.

```
      INTEGER TTY,KEY,PTR
      INTEGER WORD,ENDWRD
      LOGICAL TRACE,DONE
      DATA TTY/1/,KEY/1/,PTR/2/,TRACE/.TRUE./
C
      WRITE(TTY,100)
100   FORMAT(1H ,'START OF PROGRAM')
      CALL SETTRC(TRACE,PTR)
      CALL INISTR
      CALL INIWRD
      CALL DCLSTR(WORD,30)
      CALL DCLSTR(ENDWRD,30)
C
      WRITE(TTY,200)
200   FORMAT(1H ,'TERMINATING WORD?')
      CALL GETWRD(KEY,ENDWRD)
```

```
C
      WRITE(TTY,250)
250   FORMAT(1H ,'WORD?')
      CALL GETWRD(KEY,WORD)
      CALL EQSTR(DONE,WORD,ENDWRD)
300   IF (DONE) GOTO 400
         WRITE(TTY,350)
350      FORMAT(1H ,'WORD ENTERED...')
         CALL PUTSTR(TTY,WORD)
         WRITE(TTY,250)
         CALL GETWRD(KEY,WORD)
         CALL EQSTR(DONE,WORD,ENDWRD)
         GOTO 300
400   CONTINUE
C
      WRITE(TTY,9999)
9999  FORMAT(1H ,'END OF PROGRAM')
      STOP
      END
```

Figure 14. Using the word isolator.

SORTING APPLICATION

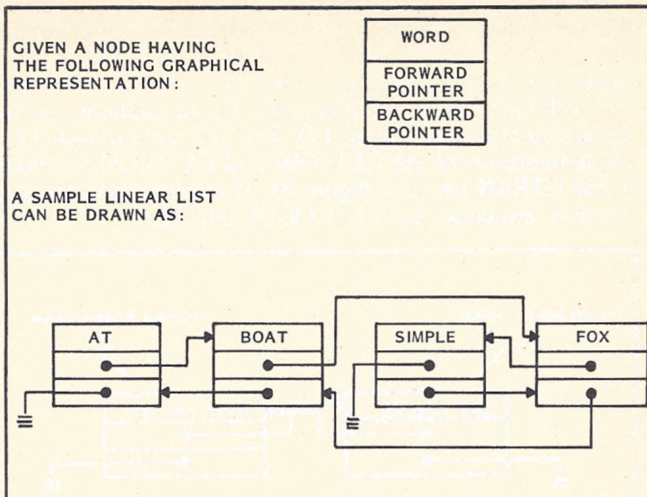
Now we can build an application to see not only how to effectively use these tools, but how new tools can be made to solve immediate and future problems. The application chosen is one which prints a sorted list of words with their frequencies of occurrence (a good use of this application is to check all words with a frequency of one for spelling errors).

One function we will need is extracting the next word from the text. Obviously, the word isolator can handle that job. Since the words are originally in unsorted order, another function must put each word in sorted sequence. Duplicate words imply incrementing a frequency counter for the duplicate word. The last required function is the printing of each sorted word with its associated frequency.

Sorting is merely the reorganization of data into some ordered form. An organization of data implies that the data has structure. This structure must, therefore, provide a way of keeping the data in sorted order. The previous chapter of this tutorial dealt with a method of sorting called the straight insertion sort. This method involved finding the correct place in a linear structure to insert the newest element so that the structure would always remain in sorted order. Recall also the Alibi concept was used in the algorithm, whereby all elements of the linear structure beyond the insertion point were physically moved to make room for the new element.

We will use this same basic algorithm and linear structure for our sort. However, instead of using an array to implement the linear structure, we will use what is called a linked list. In general, a linked list is a logical organization of individual items of data composed of numerous items called nodes, each node having several fields. The organization is called *logical* because, instead of each node being physically next to each other (like an array), it is referenced and organized by pointers (sometimes called links). Nodes need not be in physical order to be in sorted order.

For our application, each node will contain a word field and a frequency count field. While there are more complex data structures, we have chosen a simple one to introduce linked concepts: line them up in a single list or queue. In fact, the linear linked list is a pretty efficient structure when used in conjunction with the sort by insertion algorithm. For our application, each node will be connected to other nodes via a forward pointer and a backward pointer. Thus, each node can immediately reference (point to) the previous or next nodes in the list. This is done to completely separate the logic of scanning from the logic of inserting. When doing any work with linked lists, it is always advisable to draw a picture to help better understand what the structure looks like (figure 15).



NOTE: The symbol ⏚ denotes a null pointer specifying the end of the list.

Figure 15. Sample linear list of sorted words.

After seeing the form of the data structure, three conclusions regarding the straight line insertion sort algorithm can be reached. First, in order to find the insertion point, the algorithm can scan through the list by following either the forward or backward pointers. Second, nodes in the list need not be physically copied to make room for a new insertion. Only pointers need changing to keep nodes in sorted order — hence an Alias (name change) operation is all that is required (figure 16).

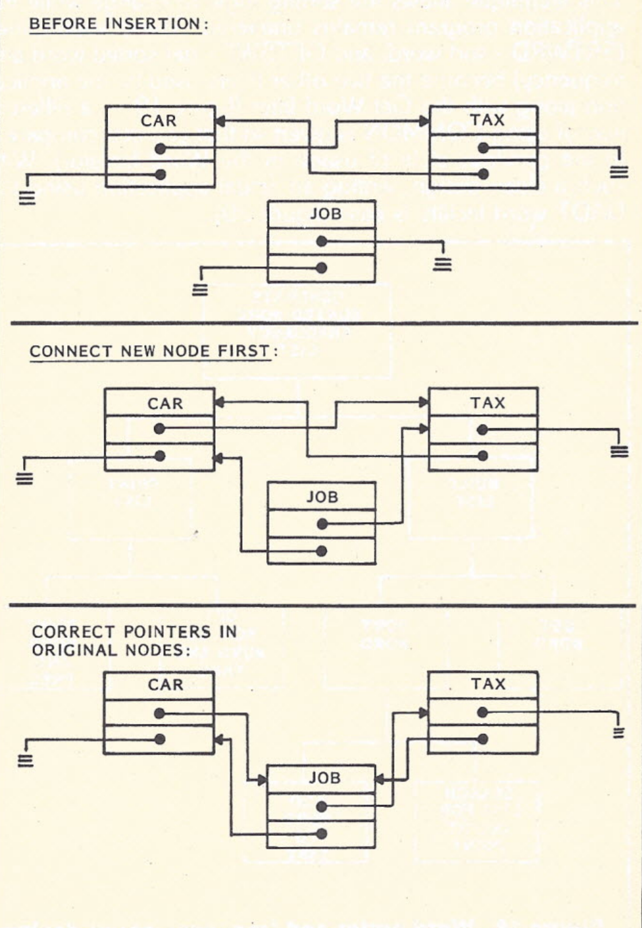


Figure 16. Inserting into list by Alias operations.

Lastly, proper initialization of the data structure can reduce the logic needed to insert a new node during special situations (e.g., when list is empty or at either end). In our case, building a list containing two nodes, one with low values and one with high values, will guarantee that all additions use the same insertion logic (figure 17). This concept prompted the implementation of the GETLOW and GETHIGH functions in the CTRAN package (figure 8). These functions are also in other languages such as COBOL and PL/I.

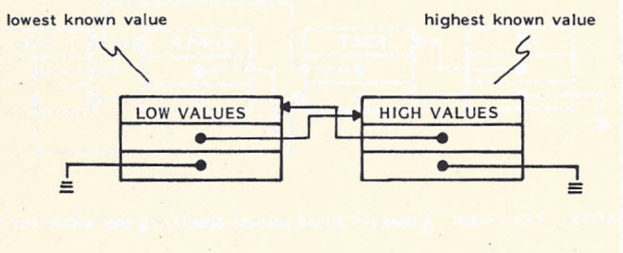


Figure 17. An initial list containing boundary nodes.

The implementation of "nodes" in the FORTRAN language is done using the RECORD data primitive. Each field in a node is represented by an array. In our case, the arrays: WRDLST, WRDFRQ, WRDFL, WRDBL hold values for word, frequency, forward pointer, and backward pointer respectively. A single INTEGER variable is used to index into the same position of each array to get field values for a node.

Knowing about all these concepts led to a general design where words were first inserted into a list, and then the complete list of words and frequencies were printed (figure 18). Separate subroutines were designed to hide the details of sorting into the list and getting sorted words from the list. This technique allows the sorting logic to change while the application program remains unaltered. These subroutines (SRTWRD - sort word, and GETSWF - get sorted word and frequency) become the two other filters used by the application along with the Get Word filter (figure 19 — a different way of using COMMON is given so that you can compare it to the previous style of usage in the Word Isolator). With such a clean design, writing an actual application using the DADT word facility is easy (figure 20).

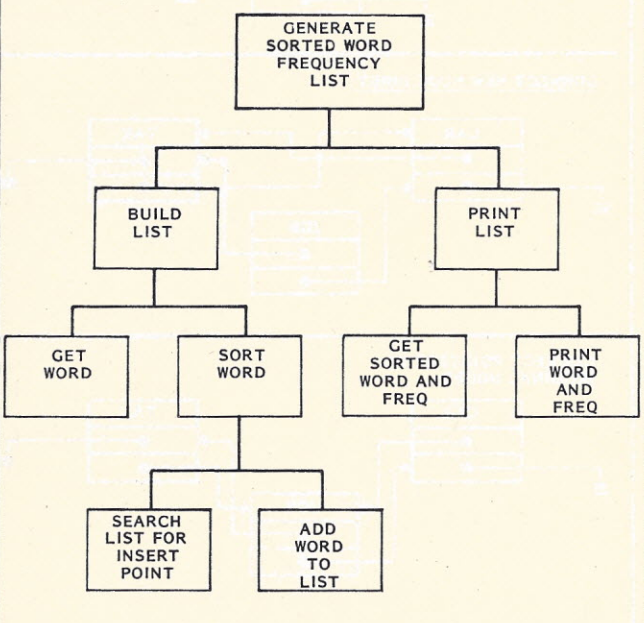


Figure 18. Word sorter and frequency count design.

```
C
C-----ADD WORD INTO SORTED WORD LIST
C
      SUBROUTINE ADDWRD(WORD,NODE)
      INTEGER WORD,NODE
      INTEGER NEWWRD,NEWNOD,NODBL
      LOGICAL SAME
      INTEGER WRDLST(100),WRDFL(100),WRDBL(100),WRDFRQ(100)
      COMMON /COMLST/WRDLST
      COMMON /COMFL/WRDFL
      COMMON /COMBL/WRDBL
      COMMON /COMFQ/WRDFRQ
      COMMON /COMNEW/NEWNOD
      CALL CALL ('ADDWRD  ')
      CALL DMPNOD(NODE)
      CALL EQSTR(SAME,WORD,WRDLST(NODE))
      IF ( NOT. SAME) GOTO 100
         WRDFRQ(NODE)=WRDFRQ(NODE)+1
         CALL DMPNOD(NODE)
         GOTO 200
100   CONTINUE
         CALL DCLSTR(NEWWRD,30)
         CALL MOVSTR(NEWWRD,WORD)
         NEWNOD=NEWNOD+1
         WRDLST(NEWNOD)=NEWWRD
         WRDBL(NEWNOD)=WRDBL(NODE)
         WRDFL(NEWNOD)=NODE
         NODBL=WRDBL(NODE)
         WRDFL(NODBL)=NEWNOD
         WRDBL(NODE)=NEWNOD
         WRDFRQ(NEWNOD)=1
         CALL DMPNOD(NEWNOD)
200   CONTINUE
      CALL RETR('ADDWRD  ')
      RETURN
      END
C
C-----BUILD SORTED WORD LIST
C
      SUBROUTINE BLDLST(DEVICE)
      INTEGER DEVICE
      INTEGER WORD,ENDWRD
      LOGICAL DONE
      CALL CALL('BLDLST  ')
      CALL DCLSTR(WORD,30)
      CALL DCLSTR(ENDWRD,30)
      WRITE(DEVICE,100)
100   FORMAT(1H ,'TERMINATING WORD?')
      CALL GETWRD(DEVICE,ENDWRD)
      WRITE(DEVICE,150)
150   FORMAT(1H ,'WORD?')
      CALL GETWRD(DEVICE,WORD)
      CALL EQSTR(DONE,WORD,ENDWRD)
200   IF (DONE) GOTO 300
         CALL SRTWRD(WORD)
         WRITE(DEVICE,150)
         CALL GETWRD(DEVICE,WORD)
         CALL EQSTR(DONE,WORD,ENDWRD)
         GOTO 200
300   CONTINUE
      CALL RETR('BLDLST  ')
      RETURN
      END
C
C-----GET SORTED WORD AND FREQ
C
      SUBROUTINE GETSWF(WORD,FREQ,DONE)
      INTEGER WORD,FREQ
      LOGICAL DONE
      INTEGER LOWNOD,HGHNOD,SRTNOD
      INTEGER WRDLST(100),WRDFRQ(100),WRDFL(100)
      COMMON /COMLST/WRDLST
      COMMON /COMFRQ/WRDFRQ
      COMMON /COMFL/WRDFL
      COMMON /COMLHN/LOWNOD,HGHNOD
      COMMON /COMSN/SRTNOD
      CALL CALL('GETSWF  ')
      IF (SRTNOD .EQ. LOWNOD) SRTNOD=WRDFL(SRTNOD)
      IF (.NOT.(SRTNOD .EQ. HGHNOD)) GOTO 100
         DONE=.TRUE.
         GOTO 200
100   CONTINUE
         CALL DMPNOD(SRTNOD)
         WORD=WRDLST(SRTNOD)
         FREQ=WRDFRQ(SRTNOD)
         DONE=.FALSE.
         SRTNOD=WRDFL(SRTNOD)
200   CONTINUE
      CALL RETR('GETSWF  ')
      RETURN
      END
```

```
C
C-----INIT SORT WORD PACKAGE
C
      SUBROUTINE INISRT
      INTEGER LOWWRD, HGHWRD
      INTEGER LOWNOD, HGHNOD, NEWNOD, SRTNOD
      INTEGER WRDLST(100), WRDFRQ(100), WRDBL(100), WRDFL(100)
      COMMON /COMLST/WRDLST
      COMMON /COMFRQ/WRDFRQ
      COMMON /COMBL/WRDBL
      COMMON /COMFL/WRDFL
      COMMON /COMLHN/LOWNOD, HGHNOD
      COMMON /COMNEW/NEWNOD
      COMMON /COMSN/SRTNOD
      CALL CALL('INISRT  ')
      LOWNOD=1
      HGHNOD=2
      NEWNOD=HGHNOD
      SRTNOD=LOWNOD
      CALL DCLSTR(LOWWRD, 30)
      CALL GETLOW(LOWWRD)
      WRDLST(LOWNOD)=LOWWRD
      WRDFRQ(LOWNOD)=0
      WRDFL(LOWNOD)=HGHNOD
      WRDBL(LOWNOD)=0
      CALL DCLSTR(HGHWRD, 30)
      CALL GETHGH(HGHWRD)
      WRDLST(HGHNOD)=HGHWRD
      WRDFRQ(HGHNOD)=0
      WRDFL(HGHNOD)=0
      WRDBL(HGHNOD)=LOWNOD
      CALL RETR('INISRT  ')
      RETURN
      END
C
C-----PRINT SORTED WORD LIST
C
      SUBROUTINE PRTLST(DEVICE)
      INTEGER DEVICE
      INTEGER WORD, FREQ
      LOGICAL DONE
      CALL CALL('PRTLST  ')
      CALL GETSWF(WORD, FREQ, DONE)
100   IF (DONE) GOTO 300
        CALL PUTSTR(DEVICE, WORD)
        WRITE(DEVICE, 200) FREQ
200     FORMAT(1H , I5)
        CALL GETSWF(WORD, FREQ, DONE)
        GOTO 100
300   CONTINUE
      CALL RETR('PRTLST  ')
      RETURN
      END
C
C-----SEARCH SORTED WORD LIST
C
      SUBROUTINE SCHLST(WORD, NODE)
      INTEGER WORD, NODE
      INTEGER LSTWRD
      LOGICAL EQUAL, GREATR
      INTEGER LOWNOD, HGHNOD
      INTEGER WRDLST(100), WRDFL(100)
      COMMON /COMLST/WRDLST
      COMMON /COMLHN/LOWNOD, HGHNOD
      COMMON /COMFL/WRDFL
      CALL CALL('SCHLST  ')
      NODE=LOWNOD
      CALL DMPNOD(NODE)
      LSTWRD=WRDLST(NODE)
      CALL EQSTR(EQUAL, WORD, LSTWRD)
      CALL GTSTR(GREATR, LSTWRD, WORD)
100   IF (EQUAL .OR. GREATR) GOTO 200
        NODE=WRDFL(NODE)
        CALL DMPNOD(NODE)
        LSTWRD=WRDLST(NODE)
        CALL EQSTR(EQUAL, WORD, LSTWRD)
        CALL GTSTR(GREATR, LSTWRD, WORD)
        GOTO 100
200   CONTINUE
      CALL RETR('SCHLST  ')
      RETURN
      END
C
C-----SORT WORD
C
      SUBROUTINE SRTWRD(WORD)
      INTEGER WORD
      INTEGER INSNOD
      CALL CALL('SRTWRD  ')
      CALL SCHLST(WORD, INSNOD)
      CALL ADDWRD(WORD, INSNOD)
      CALL RETR('SRTWRD  ')
      RETURN
      END
C
      SUBROUTINE DMPNOD(NODE)
      INTEGER NODE
      INTEGER WRDLST(100), WRDFRQ(100), WRDBL(100), WRDFL(100)
      COMMON /COMLST/WRDLST
      COMMON /COMFRQ/WRDFRQ
      COMMON /COMBL/WRDBL
      COMMON /COMFL/WRDFL
      CALL CALL('DMPNOD  ')
      WRITE(2, 1)NODE
1     FORMAT(1H , 'NODE=', I8)
      WRITE(2, 2)WRDLST(NODE)
2     FORMAT(1H , 'WORD=', I8)
      WRITE(2, 3)WRDFRQ(NODE)
3     FORMAT(1H , 'FREQ=', I8)
      WRITE(2, 4)WRDFL(NODE)
4     FORMAT(1H , 'FL=', I8)
      WRITE(2, 5)WRDBL(NODE)
5     FORMAT(1H , 'BL=', I8)
      CALL RETR('DMPNOD  ')
      RETURN
      END
```

Figure 19. Word sorter and frequency count code.

```
      INTEGER TTY, KEY, PTR
      LOGICAL TRACE
      DATA TTY/1/, KEY/1/, TRACE/.TRUE./, PTR/2/
      WRITE(TTY, 100)
100   FORMAT(1H , 'START OF PROGRAM')
      CALL SETTRC(TRACE, PTR)
      CALL INISTR
      CALL INIWRD
      CALL INISRT
      CALL BLDLST(KEY)
      CALL PRTLST(TTY)
      WRITE(TTY, 9999)
9999  FORMAT(1H , 'END OF PROGRAM')
      STOP
      END
```

Figure 20. Using the word sorter.

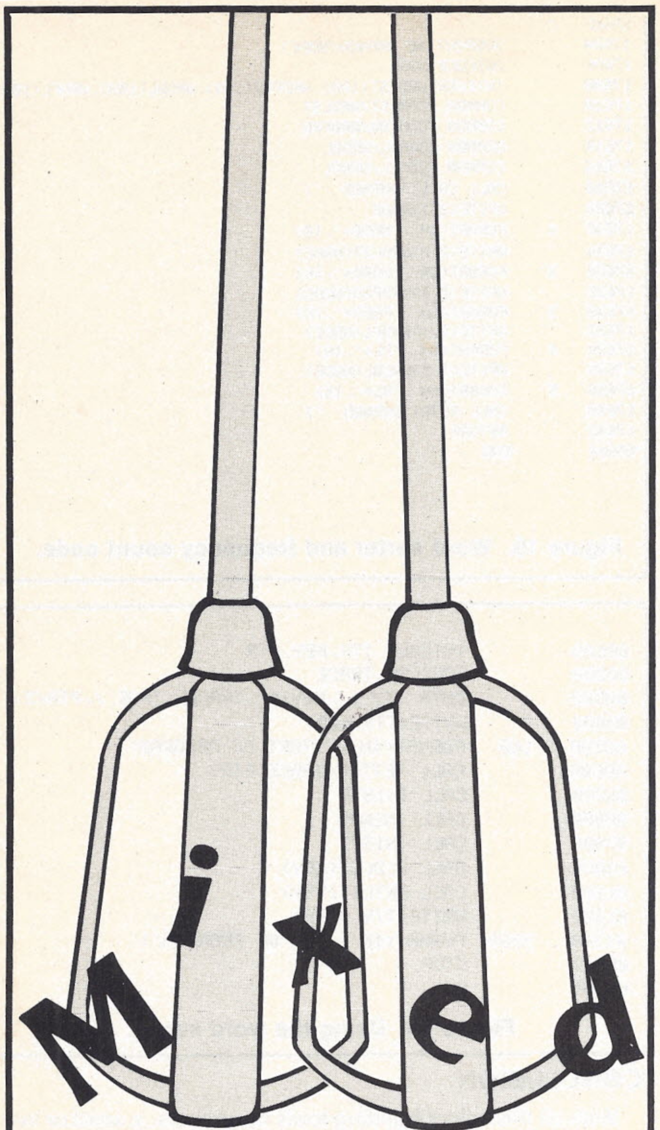
CONCLUSION

With all this talk of building tools upon tools, a word or two of caution is in order. One can spend all one's time building tools, and not solving problems. The microcomputer should be a help to the small business, but it cannot if only tools are built and not applications. Thus careful assessment must be made before plunging into the construction of a new software tool.

Another common area of trouble is the proper selection and use of software utilities when your toolbox is large. Having a large set of software capabilities can often lead one into choosing a bad approach to implementing a problem solution. A well-documented and complete understanding of what is available will help you choose the right tool for the right job.

Experience in constructing these tools has pointed out that the microcomputer environment is still somewhat unpredictable. Several compiler or run-time limitations caused long delays in testing by having to track down difficult bugs. The trace facility made the identification of those problem areas much easier, however. It is recommended that the trace tool be given the power to display input and output parameters of the called subprograms. This additional feature was the biggest help during testing.

Probably the major advantage in understanding the software tool concept is that one can stop getting caught up in the "invented here syndrome." Most of the software application shops today do not really build upon the work of others. In fact, most applications are still built over and over again from scratch because past work is not in a reusable form. Hopefully, this article will be a start towards the building of better engineered application software in the micro-environment.□



Interfaces

By Richard A. Leary

While each major microprocessor has one or more LSI peripheral interface devices specifically designed for that processor, occasions may arise when those "matched" hardware blocks do not satisfy the user's needs. It is also possible that changes to existing equipment may force one into mixing devices of one family with those of another. As a result of the latter kinds of pressure, I found myself in a position where I had to interface I/O devices of the 6800/6502 family to a Z-80. What follows is both a synopsis of my problems and a description of what I did to solve those problems.

The 6800/6502 devices I used were part of wire-wrapped I/O board I had built over a year earlier to use with a wire-wrapped 6502 CPU board. In switching to a Z-80 CPU (primarily due to the birth of 6502 software) the problem was how to make the transition with minimum trouble and expense. While both the CPU and the I/O board used S-100 prototype boards and the rest of my system used the S-100 bus, that fact alone does not guarantee compatibility.

Although I thought I had carefully checked the I/O board and the rest of the system for compatibility with the Z-80 CPU board, the problems started building up early. When I first attempted to run my system with the Z-80 CPU in place, nothing seemed to happen. My initial reaction, since I do not have a front panel and thus rely upon a monitor in ROM to bring the system up, was that somehow my 1K hand assembled monitor, hand burned into a 2708, was in error. While there was certainly ample opportunity for that process to have gone astray, that proved not to be the case. Although I suspected the CPU-I/O interface from the start, I spent many hours investigating the possibility of a ROM error.

At this point, it is well to back up a minute and look at how the 6502 was originally interfaced to the I/O devices. Figure 1 shows an example of the interface in semi-block diagram form. Note that while the key interface signals are shown, I have not shown all the on-board decoding and buffer enabling logic nor have I shown other signals which are not germaine to this discussion.

As can be seen, I had defined a new bus line (66) as "I/O." This line was simply the output of a comparator on the CPU board which was true whenever a preselected page in memory was addressed. As the 6502 and 6800 have no I/O instructions like the 8080 or Z-80, this preselected page defined the range of I/O addresses to be used in my system. The "I/O" signal fed all I/O devices attached to the bus. What that meant was that any I/O device need only look for the "I/O" signal and then decode the 8-bit port number much as is done for the 8080 or Z-80. This demonstrated some measure of S-100 compatibility for I/O instructions.

To switch to the CPU I changed the 6502 interface to what is shown in Figure 2. Since I wanted to retain the ability to interface both CPUs with my system merely by switching CPU boards, this dual interface was implemented with some logic on the I/O board which sensed which CPU board was being used and set a switch accordingly. The insides of that switch will not be discussed in detail since it was a simple piece of selection logic. Its sole purpose was to determine what signal would be used for the ENABLE signal going to the 6820. As before, for the 6502, that signal was 02. In Figure 2 it is clear that the only real change, other than the introduction of the switch discussed above, is in how the CS (chip select) signal for the I/O device is generated.

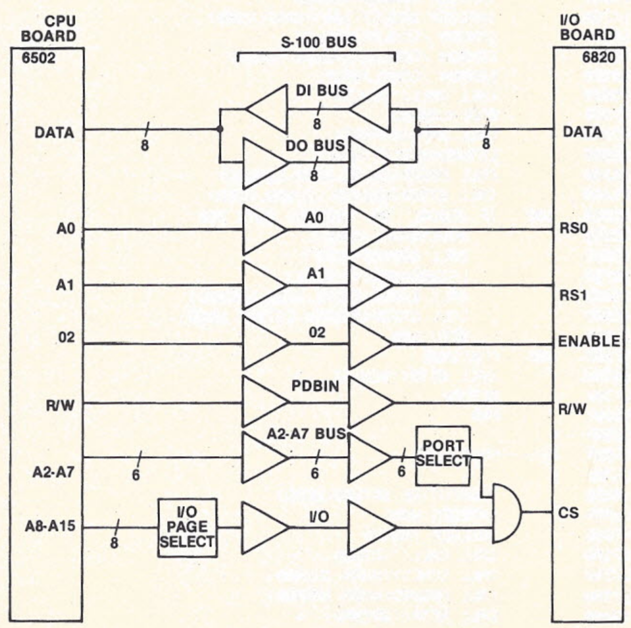


Figure 1. Original 6502-6820 Interface.

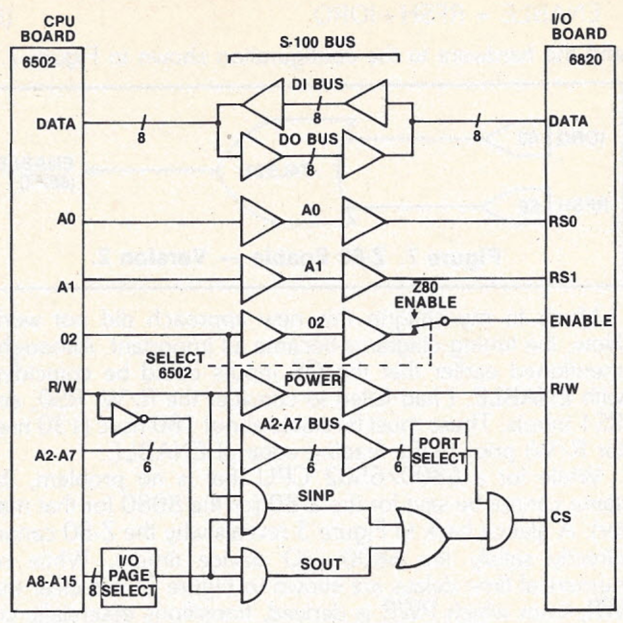


Figure 2. Modified 6502-6820 interface.

While in Figure 1

CS = PORTSELECT•I/O (1)

in Figure 2

CS = PORTSELECT•(SINP+SOUT) (2)

However, if the terms SINP and SOUT in equation (2) are decomposed, equation (2) really says

CS = PORTSELECT•([I/O•R/W]+[I/O•$\overline{\text{R/W}}$•02]) (3)

Ignoring that extra 02 term in this last equation (3), it is clear that equation (1) and equation (2) are equivalent. Why the 02 term can be ignored is not immediately obvious but is the result of the 6800/6502 I/O device family characteristics. While the CS inputs must be stable during the ENABLE period, they do not have to be stable prior to that time. As ENABLE in this case is 02, the fact that one of the factors in determining CS involves an AND with 02 means that the I/O device in Figure 2 should work just as it did in Figure 1. In actual fact that is what happens.

Then why change? The answer is simple; an S-100 Z-80 CPU generates SINP and SOUT, not I/O. So this change

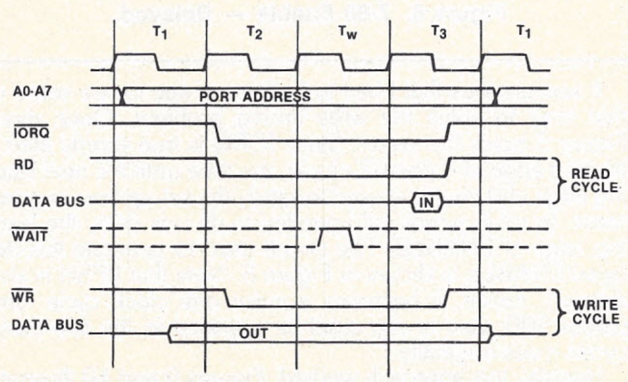


Figure 3. Z-80 I/O Cycles.

was one that I thought prudent in order to make the system "S-100 compatible."

The next question was what should be used for the ENABLE signal when operating with the Z-80 CPU. My initial hope that I could continue to use 02, now the Z-80 0 clock, was quickly put to rest without having to reach the bread board stage. The 0 signal (shown on Figure 3 which was extracted from Zilog Z-80 CPU Technical Manual) which for my CPU, an Ithaca Audio board, becomes the S-100 02 (24) obviously does not satisfy the ENABLE timing requirements shown in Figures 4a and 4b. The later two figures were extracted from the Motorola M6820 data sheet and are typical of all 6800 family devices. What was needed was a signal that matched the Z-80 CPU's actual read and write timing. Since SINP and SOUT are derived in the Z-80 CPU board as

SOUT = IORQ•WR (4)

and

SINP = IORQ•RD (5)

these two signals appeared to be the ideal candidates.

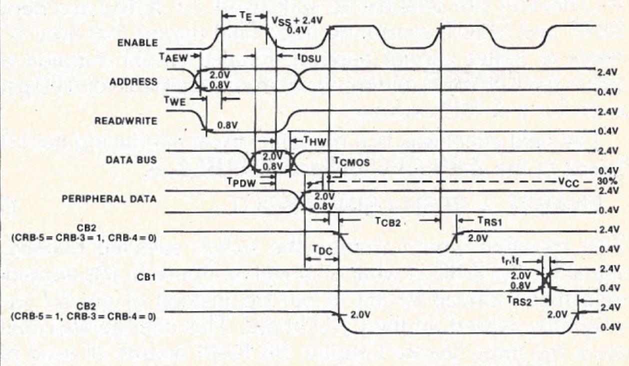


Figure 4a. 6820 Write Cycle.

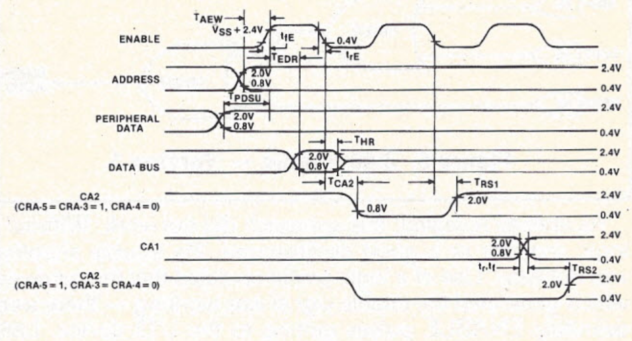


Figure 4b. 6820 Read Cycle.

However, one other requirement of the I/O devices had to be satisfied. The ENABLE signal, as shown in Figure 4, had to continue to clock-like fashion even after the device was addressed in order for the peripheral handshaking and interrupt response functions to work correctly. If enable only consisted of the following:

ENABLE = SINP + SOUT (6)

it is clear that this last requirement would not be met.

One feature of the Z-80 offered some promise. Unlike either the 8080, 6800, or 6502, the Z-80 generates a special signal, $\overline{\text{RFSH}}$, designed to be used to refresh dynamic memories. As Figure 5 indicates, this signal has some interesting characteristics which meant it might satisfy the requirements discussed previously. First, it occurs during every M1, i.e. OP-Code fetch, cycle. That means that while not occurring

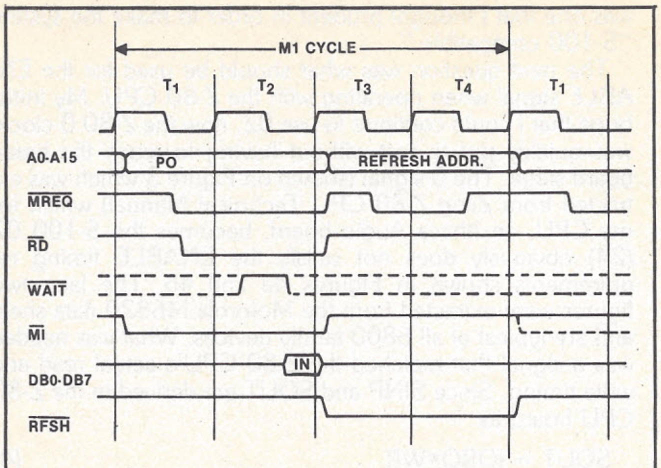


Figure 5. Z-80 Op Code Fetch Cycle.

nearly as often as a 6800/6502 02 clock, it still occurs often enough to have negligible effect on most low speed I/O operations. Second, it occurs during a time when no I/O devices can possibly be addressed, i.e. IORQ and hence SINP and SOUT cannot be true. That means that false I/O reads or writes cannot happen. Finally, $\overline{RFSH}$ duration exceeds the ENABLE minimum width requirements of 470 μsec even for a 4 MHz system.

The next effect was that my initial try at interfacing the I/O board to the Z-80 CPU defined ENABLE as

$$\text{ENABLE} = \text{RFSH} + \text{SINP} + \text{SOUT} \tag{7}$$

This equation would change the switch selected ENABLE signal for the Z-80 to what is shown in Figure 6. (As an aside, note that I moved RFSH to pin 66 instead of pin 67 as it originally existed on the CPU board. This change was made since my three Vector Graphic 8K RAM boards all used pin 67 as $\overline{\text{PHANTOM}}$.)

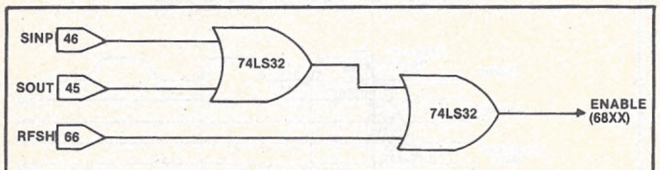


Figure 6. Z-80 Enable — Version 1.

As already revealed, this approach did not work. Without a logic analyzer or a good oscilloscope, the trouble shooting was not easy. Use of a logic probe revealed that the hardware which generated the enable signal was working — there were definitely ENABLE pulses getting to the I/O device. Little else was obvious as a result of a lot of testing.

The problem was that there were two major mistakes in my design. Neither of the problems was found by test equipment but rather by long hours looking at the CPU board schematic and the Z-80 and 6800 timing diagrams.

First, the schematic related problem. Remember that I had said earlier that I thought my 6502 CPU board and hence the I/O board were "S-100 compatible." This wasn't true. The key discrepency was in failing to remember that the S-100 bus latches SINP and SOUT until the next I/O or memory cycle. That meant that the Z-80 was enabling the I/O devices long after they should have been in order to satisfy the 6800 timing.

The obvious cure was to find an unlatched signal. Obviously, the signal I needed was IORQ itself which unfortunately does not appear on the S-100 bus. Since I had already departed from the ideal by using RFSH, the use of IORQ (routed to bus pin 63) did not seem out of place. After all the objective was *to make it work,* not necessarily be "S-100 compatible" (whatever that really means). That changed the logic to

$$\text{ENABLE} = \text{RFSH} + \text{IORQ} \tag{8}$$

and the hardware to the configuration shown in Figure 7.

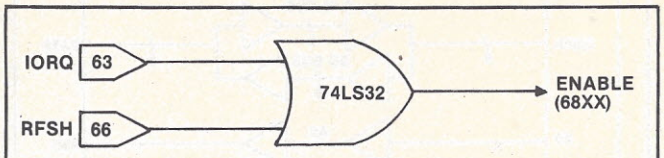


Figure 7. Z-80 Enable — Version 2.

Much to my chagrin this new approach did not work. Now, the timing diagrams became all important. Although I mentioned earlier that the CS inputs could be coincident with ENABLE, I had failed to consider the R/W, RS0, and RS1 inputs. These must be stable about 180 nsec (130 nsec for R/W) prior to the leading edge of ENABLE.

While for a 6800/6502 CPU that is no problem, the same cannot be said for the Z-80 (or the 8080 for that matter). A glance back to Figure 3 reveals why the Z-80 cannot directly satisfy the 6800 I/O device timing. While no numerical time delays are shown in Figure 3, it is clear that $\overline{\text{WR}}$, from which $\overline{\text{PWR}}$ is derived, transitions essentially coincident with $\overline{\text{IORQ}}$. $\overline{\text{IORQ}}$ is, of course, what I had most recently used to generate ENABLE. Note, there is no such problem during read operations as long as R/W is the same as $\overline{\text{WR}}$. It will be at the READ level all the time, at least until a write I/O operation comes along. Note, also in Figure 3 that as long as RS0 and RS1 are derived from the port address, i.e. SINP, $\overline{\text{PWR}}$ or similar signals are *not* used, they will meet both read and write cycle timing requirements.

It was apparent that while I could read the I/O devices I could not write to them. It was at this point that a peculiarity of the Z-80 came to the rescue. Note again in Figure 3 that the Z-80 automatically adds a wait state into the middle of all I/O operations. The net effect of that is that IORQ is true for something like 1.250 μsec for a 2 MHz clock. That is over twice as long as the standard speed 6800 I/O devices required for ENABLE.

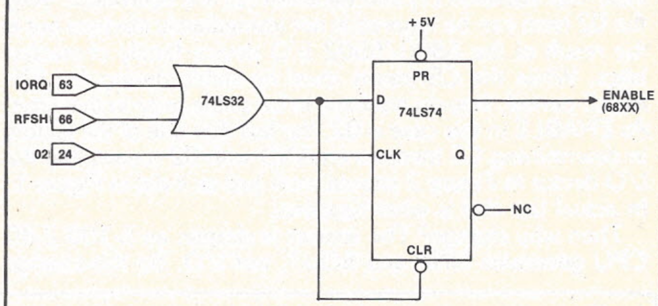


Figure 8. Z-80 Enable — Delayed.

It was obvious that I had to find some way to use some of that time to solve the write timing problem. Once more Figure 3 holds the secret. Since IORQ is true shortly *after* a leading edge of 0 (the 02 clock), the time until the next leading edge of 0 could be used to satisfy the set-up time requirement. In other words, all I needed to do was delay the leading edge of ENABLE. The circuit used to generate this delayed ENABLE is shown in Figure 8. Note that RFSH is also delayed, hence, it becomes a pulse one clock cycle long (about 500 μsec in my case) as opposed to the two clock cycles it was originally.

Happily, this approach worked. Figures 9 and 10 illustrate the timing relationships between the new ENABLE signal and the other CPU generated signals. In each figure the period in which ENABLE is true has been indicated by cross hatching on either the $\overline{\text{IORQ}}$ or $\overline{\text{RFSH}}$ line.

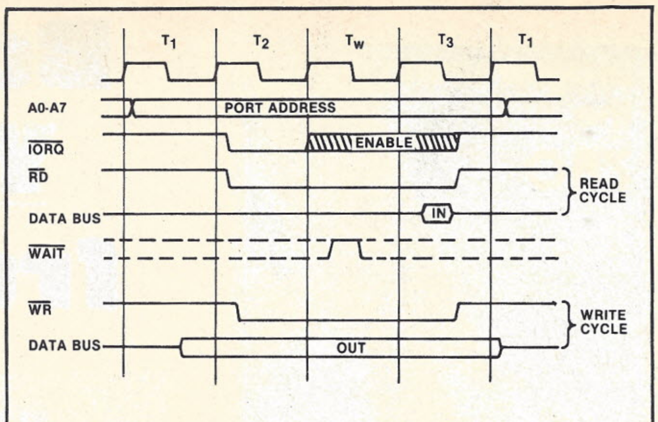


Figure 9. Z-80 I/O Cycles — Enable.

The key question remaining is whether or not extrapolation of these results to other system configurations is possible. The answer is a clear yes. The actual approach taken may differ but the general approach is the same — use the CPU generated signals and the system clock to create the necessary delays. For example, consider that even though the 8080 does not automatically generate that valuable extra wait state during I/O operations, one or more could easily be generated by the I/O board logic and then used to create the necessary I/O device timing in conjunction with the 01 or 02 clock. Similarly an "unlatched IORQ" could be generated using SINP and SOUT again synchronized to the 01 and 02 clocks. And of course a RFSH-like signal could be synthesized using the M1 status signal and the clocks.

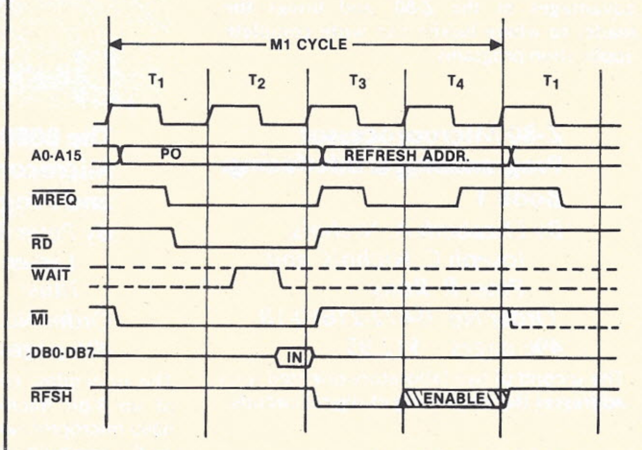


Figure 10. Z-80 Op Code Fetch Cycle.

What to do about a 2.5 MHz or 4 MHz Z-80 is also clear. For a 2.5 MHz Z-80 the IORQ signal should be delayed but the RFSH signal should not be. At 2.5 MHz RFSH is roughly 800 ns long, so the delay would cut it to about 400 ns — just a bit too short. All that means is that the OR should be after the delaying flip-flop rather than before as was done for the 2 MHz case.

At 4 MHz things get a bit messy. RFSH is now 500 μsec so it can still be used directly but IORQ is short. One could, of course, use one of the higher speed devices that Motorola and others now market but the premium paid for those devices may be too steep. Probably the best answer is to add a wait state for the I/O cycles.

One aspect of all this discussion should be clear by now. The key to successfully interfacing devices of different families is a thorough understanding of the timing needs at both ends of the interface. In addition, the secret to implementing that interface is full and complete utilization of the available signals, especially the system clock.□

I made the TRS-80* into a serious computer. Now I've made the Model II into a spectacular one.

I'm Irwin Taranto, and I've helped almost a thousand businesses get their first computers up and running.

I've done it primarily with the TRS-80, because it's a really elegant piece of hardware. Given the right programs, it can do substantially the same work as the traditional minicomputers that cost four times as much.

I proved it with four on-line, interactive programs adapted from the genuine Osborne & Associates systems, originally designed for the $30,000 Wang computer. Then I added two of my own and made them all work on a $4000 TRS-80.

Now I've done the same thing for the new TRS-80 Model II. It's an $8000 computer that works twice as fast and has four times the memory—up to two million characters.

My new systems are fully documented, and because I'm working with a much more powerful computer, they're a night-and-day advance over the Model I programs. They'll turn your Model II into a complete business computer, set up and ready to go.

THE TRS-80 MODEL II PROGRAMS

General Ledger/Cash Journal: handles up to 7000 transactions on 500 different user-defined accounts. It keeps track of them by month, quarter and year, makes comparisons to the prior year, and does departmentalization.

Accounts Payable/Purchase Order: generates the purchase order and posts the item to payables when the goods are received. Invoice-linked, it calculates and prints checks and aged ledger reports and links fully to the general ledger.

Accounts Receivable/Invoicing: keeps track of billed and unbilled invoices, open and closed items, aging and service charge calculation. It prints statements, links to the general ledger, and can work within either an invoice-linked or balance-forward accounting system.

Payroll/Job Costing: computes regular, overtime and piecework pay, keeps employee files, figures taxes and deductions, prints checks, journal, 941-A and W-2 forms, and breaks out individual job costs.

*A trademark of the Tandy Corporation.

When I say set up and ready to go, I mean just that. If you're not quite sure on that point, call the number below and we'll give you the names of some of the people who've already bought all over the world. Call them up and hear what they have to say.

These Model II programs are completely custom-tailored, which explains their $249.95 price. Before we'll send you a disk, you have to fill out a detailed questionnaire that tells us your precise business requirements. Then we send you the disk, all the instructions you need, and my phone number. If you call, we answer all your questions. If your questions are tough enough, I'll talk to you personally.

Because that way I'll make sure that Model II of yours turns into a spectacular computer, just like I promised.

☐ Please send me the custom questionnaires for the following $249.95 Model II programs:

- ☐ General Ledger/Cash Journal
- ☐ Accounts Payable/Purchase Order
- ☐ Accounts Receivable/Invoicing
- ☐ Payroll/Job Costing

☐ Please send me information on the TRS-80 Model I programs at $99.95 each

☐ Please send me information on other Taranto business programs

Your name ______________________

Company name ______________________

Address ______________________

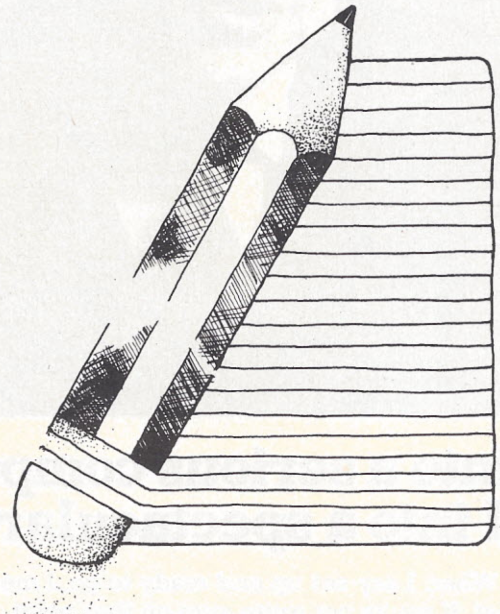
City/State/Zip ______________________

Taranto
& ASSOCIATES, INC.

P.O. Box 6073, 4136 Redwood Hwy., San Rafael CA 94903 • (415) 472-2670

An Error Correcting Memory

By Terry Dollhoff



INTRODUCTION

The use of a parity detection circuit to detect memory read errors is becoming popular with the hobbyist seeking more reliable memory operation. If the hardware can detect an error in memory, then it can also correct that error. This article describes the implementation of a single error correcting, double error detecting memory for the Technico 16-bit TI9900 based system. The Technico system was chosen to illustrate the hardware algorithm because error correction is more efficient with a 16-bit data word than with an 8-bit one. The hardware described here is now in operation with the 9900. A dramatic demonstration of the hardware is to remove *any* one memory chip and observe that the software is unaffected.

MATHEMATICAL OUTLINE

The best place to begin is with a brief summary of the mathematical development of the error correcting code. It is not intended as a comprehensive development of error correcting codes, but it is an overview of the specific code used for one bit error correction, and two bit error detection of a 16-bit memory.

The error correcting code used for single error correct double error detect is called a cyclic code. All cyclic codes are based upon the concept of a primitive polynomial. The definition is not really important here; just assume that the polynomial presented is a primitive one. Now, if P(X) is a primitive polynomial of degree m, then it can be proved that a cyclic code produced by the polynomial:

$$B(X) = (1+X)*P(X)$$

is a double error detecting, single error correcting code. We will discuss later how the polynomial produces a code. In particular, the code produced by B(X) has the following properties:

n (total code length) $= 2^m - 1$
n−k (no. of parity bits) $= m + 1$
K (no. of information bits) $= 2^m - m - 2$

If m=5 is chosen, then B(X) will reproduce a code with the following properties:

n = 31 bits
n−k = 6 parity bits
k = 25 data bits

This code has the correction properties needed, but it has 25 data bits instead of 16. To arrive at a 16-bit code we can use a shortened cyclic code which is based upon the above code. A shortened cyclic code is one where all unused data bits are assumed to be zero, and it has the same correction properties. Thus a 16-bit code based upon the above code can be defined.

The first step in developing the code is to locate B(X). Since $(1+X^2+X^5)$ is a primitive polynomial of degree m=5, the definition for B(X) is:

$$\begin{aligned} B(X) &= (1+X)(1+X^2+X^5) \\ &= 1 + X + X^2 + X^3 + X^5 + X^6 \end{aligned}$$

Notice that the operator "+" is the addition operator for a group of two elements and is actually a binary *exclusive-or.* In fact, the above polynomial can be represented as the following binary number:

$$B(X) = 1\ 1\ 1\ 1\ 0\ 1\ 1$$

since;

$$B(X) = 1*1+1*X+1*X^2+1*X^3+0*X^4+1*X^5+1*X^6$$

If all data words are also represented as a polynomial, say M(X), then B(X) produces a code by the following equation:

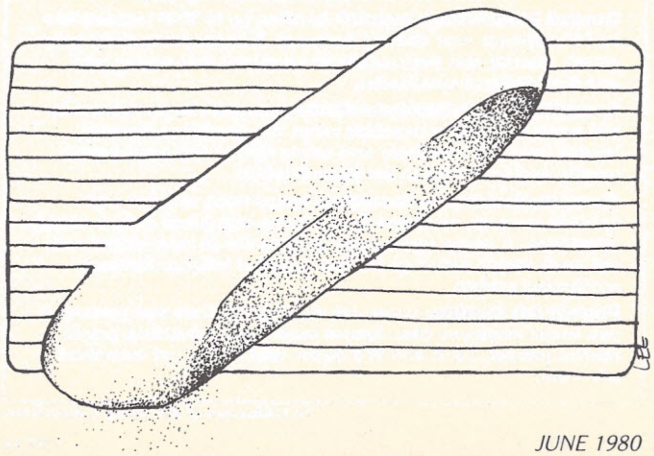
$$E(X) = B(X)*M(X)$$

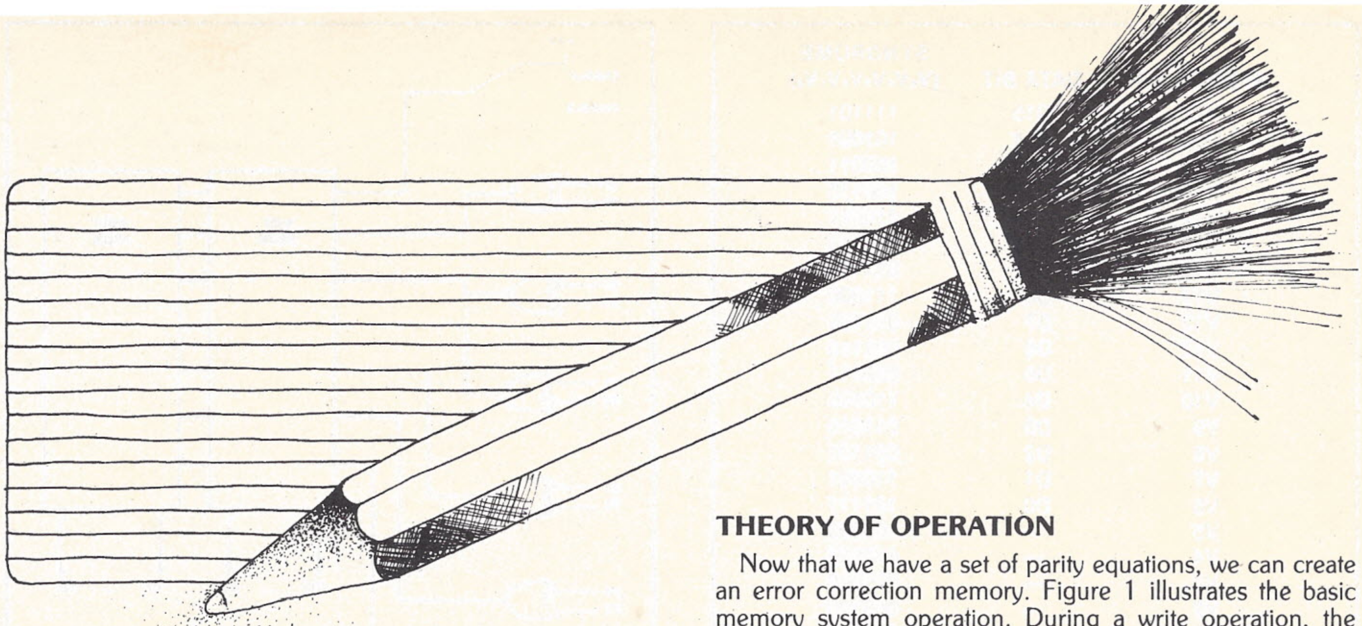
We will not investigate the use of this form of the code because it has one distinct implementation disadvantage. That is, all the stored data bits are functions of the input data bits (not just the parity bits). This means it is more difficult to test the memory because all bits are altered before being stored. This problem can be solved by changing the code to a systematic code. Then the data bits are stored unchanged and only the six parity bits are functions of the data bits. To change to a systematic code, calculate another polynomial, H(X), as:

$$H(X) = (X^n + 1)/B(X)$$

or;

$$\begin{aligned} H(X) &= (X^{31}+1)/(1+X+X^2+X^3+X^5+X^6) \\ &= 1+X+X^4+X^6+X^7+X^9+X^{11}+X^{12}+X^{14}+X^{16}+ \\ &\quad X^{20}+X^{23}+X^{24}+X^{25} \end{aligned}$$





If the incoming data bits are represented as:

$\text{Data} = V_6V_7...V_{30}$

where (V_{22} to V_{30} are zero), the encoded data is:

$$\text{Encoded} = \underbrace{V_0V_1V_2V_3V_4V_5}_{\text{parity}}\underbrace{V_6V_7...V_{21}}_{\text{data}}\underbrace{V_{22}...V_{30}}_{\text{zero}}$$

The equations for the parity bits can be derived from H(X) and are:

$$V_{n\text{-}k\text{-}j} = \sum_{i=0}^{k} h_i{*}V_{n\text{-}i\text{-}j} \quad \text{for } j=1 \text{ to } n-k$$

where;

$H(X) = h_i{*}x^{i-1}$

Since we are only interested in storing 16-bit data, V_{22} to V_{30} are all assumed to be zero. With this assumption in mind, the parity equations can be expanded as:

$$\begin{aligned}
V_5 &= V_{21}+V_{20}+V_{19}+V_{18}+V_{16}+V_{14}+V_{10}+V_7+V_6 \\
V_4 &= V_{20}+V_{19}+V_{18}+V_{17}+V_{15}+V_{13}+V_9+V_6+V_5 \\
V_3 &= V_{21}+V_{19}+V_{18}+V_{17}+V_{16}+V_{14}+V_{12}+V_8+V_5+V_4 \\
V_2 &= V_{21}+V_{20}+V_{18}+V_{17}+V_{16}+V_{15}+V_{13}+V_{11}+V_7+V_4+V_3 \\
V_1 &= V_{20}+V_{19}+V_{17}+V_{16}+V_{15}+V_{14}+V_{12}+V_{10}+V_6+V_3+V_2 \\
V_0 &= V_{21}+V_{19}+V_{18}+V_{16}+V_{15}+V_{14}+V_{13}+V_{11}+V_9+V_5+V_2+V_1
\end{aligned}$$

Certain of the parity bits are based upon other parity bits. This interaction would cause an implementation problem and the interactions can be removed by expanding the equations as illustrated below (as a shorthand notation, write 21 for V_{21}, etc.).

$$\begin{aligned}
V_4 = {} & 20+19+18+17+15+13+9+6+ \\
& 21+20+19+18+16+14+10+7+6
\end{aligned}$$

Since the + operator is an exclusive-or, $X+X=0$. Therefore the above equation can be reduced to:

$$V_4 = 21+17+16+15+14+13+10+9+7$$

Expanding the other parity equations in a like manner, we can calculate the final set of parity equations.

$$\begin{aligned}
V_5 &= 21+20+19+18+16+14+10+7+6 \\
V_4 &= 21+17+16+15+14+13+10+9+7 \\
V_3 &= 21+20+16+15+14+13+12+9+8+6 \\
V_2 &= 21+18+16+15+13+12+11+10+8+6 \\
V_1 &= 19+18+17+16+15+12+11+9+6 \\
V_0 &= 21+20+19+17+15+11+8+7+6
\end{aligned}$$

THEORY OF OPERATION

Now that we have a set of parity equations, we can create an error correction memory. Figure 1 illustrates the basic memory system operation. During a write operation, the parity circuit calculates the six parity bits which are then stored with the original data bits. During a read operation, the parity circuit calculates a new set of parity bits based upon the stored data bits. If this newly calculated set of parity bits is the same as the stored parity bits, there is no data error (or more than two errors since this code is limited to double error detection). If the read parity and the stored parity are not the same, the data or stored parity are in error.

To diagnose the problem, the two sets of parity bits are exclusive-ored together to produce a new value called the syndrome. This syndrome is used to correct the error (single bit error) or to detect an uncorrectable error (two bit error).

Prior to implementing the correction circuit, it must be determined which syndrome will be produced by each single bit error. This can be done empericaly. If any single bit is in error, every parity equation involving the errant bit will be reversed in value so the syndrome bit will be set. As an example, if bit 13 is in error, then parity bits V_4, V_3, V_2 are all altered. Thus the single error syndrome for bit 13 is (011100). If this syndrome is encountered during a read,

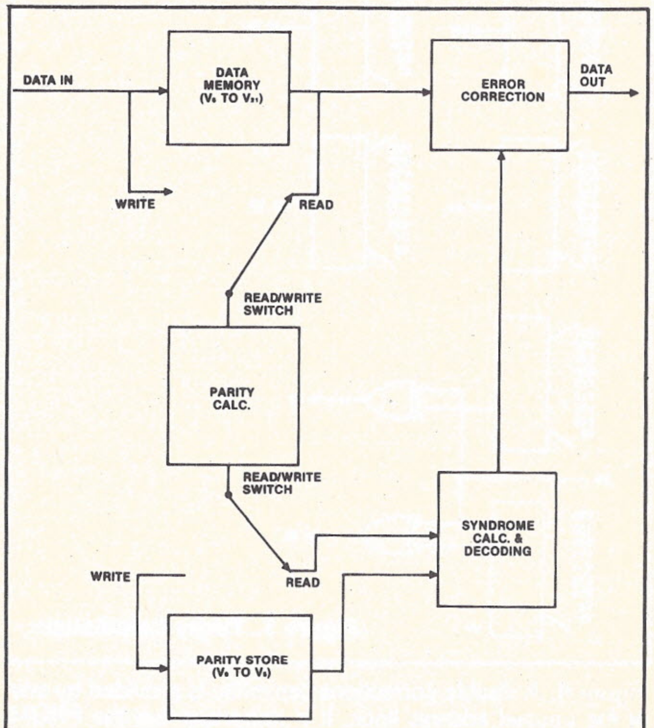


Figure 1. Error Correction Block Diagram.

STORED BIT	DATA BIT	SYNDROME ($V_5V_4V_3V_2V_1V_0$)
V21	D15	111101
V20	D14	101001
V19	D13	100011
V18	D12	100110
V17	D11	010011
V16	D10	111110
V15	D9	011111
V14	D8	111000
V13	D7	011100
V12	D6	001110
V11	D5	000111
V10	D4	110100
V9	D3	011010
V8	D2	001101
V7	D1	110001
V6	D0	101111
V5	—	100000
V4	—	010000
V3	—	001000
V2	—	000100
V1	—	000010
V0	—	000001

Figure 2. Error Syndromes.

then bit 13 is in error. To correct that bit, simply reverse it. Any syndrome not in the single bit syndrome family indicates a noncorrectable error. All of the single bit syndromes are shown in Figure 2.

IMPLEMENTATION

Four of the parity equations include nine terms and are easily calculated by a 9-bit generator (74280). The other two have ten terms and require a 9-bit parity generator and an exclusive-or gate (7486). This circuit is shown in Figure 3.

The syndrome is calculated by exclusive-or of the stored parity (called P_0 to P_5) with the new one (called V_0 to V_5). This value is used to address an error correction PROM which will yield $C_i = 1$ if bit D_i is in error. This is shown in

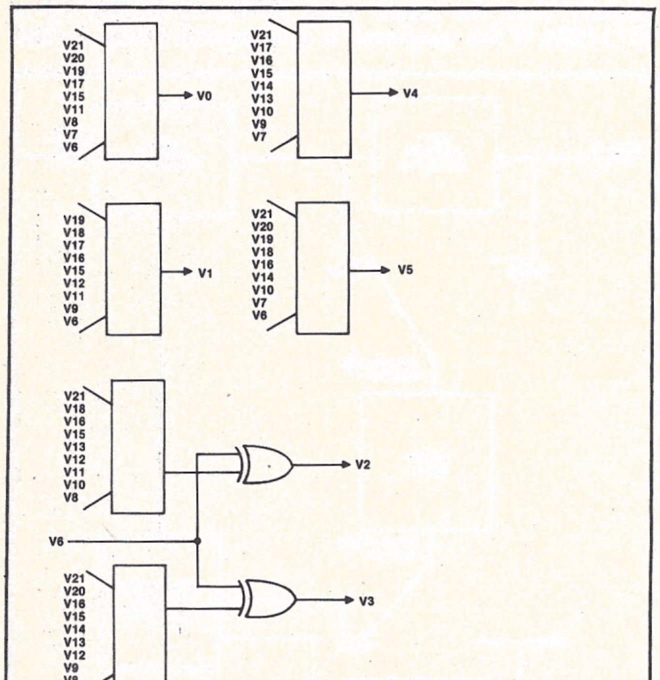


Figure 3. Parity Calculation.

Figure 4. A disable corrections capability is provided by one of the unused address lines. It is important that the PROM have a fast access time. We found the TI74S470 perfect for this application.

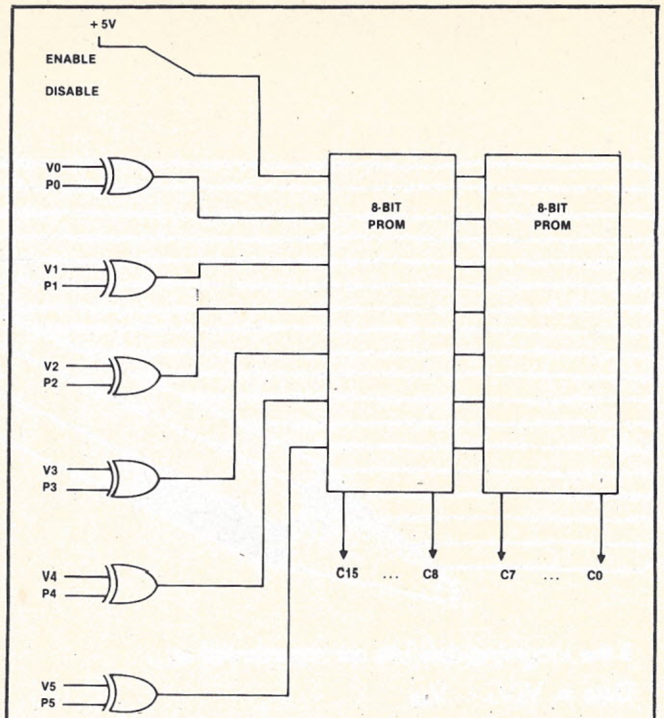


Figure 4. Syndrome Calculation and Decode.

Figure 5 shows the actual error correction. Just exclusive-or the output of the error correction PROM with the memory data. An uncorrected error will be indicated by $C_{15} = C_{14} = 1$ or any other two bits since it is a single error code.

Figure 6 shows the overall bus control. Only one extra set of buffers (for the parity bits) is required. The input data buffers are probably needed elsewhere in the system for other reasons. Note that when the parity is written, the syndrome is zero since the stored and computed parity are equal. If not zero, the buffers, memory for parity, or parity gates have failed.

The circuit in Figure 6 is idealized since we have not addressed generation of dynamic memory refresh or other specialized memory controls. Its only purpose is to illustrate the philosophy of the error correcting circuits.

During a write, the CPU data is stored directly in the Data Memory and the parity is stored in the Parity Memory. During a read, the data from memory is used to calculate a new parity (V_0-V_5). If the new parity differs from the stored one (P_0-P_5), the error bit is corrected by the error correction circuit.

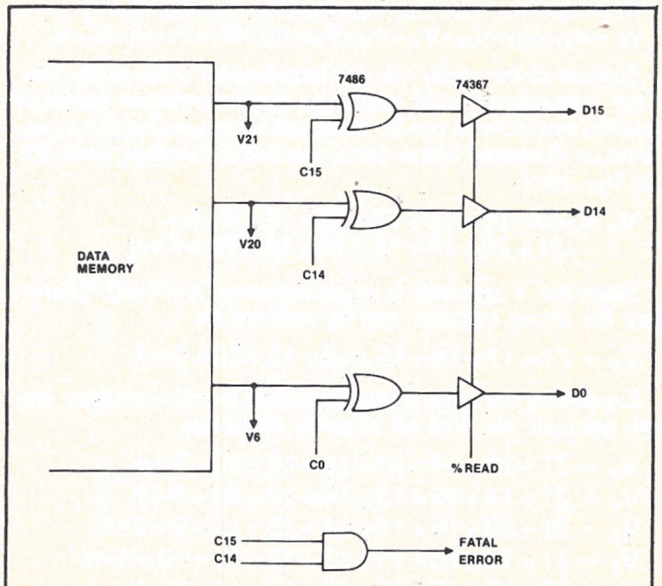


Figure 5. Error Correction.

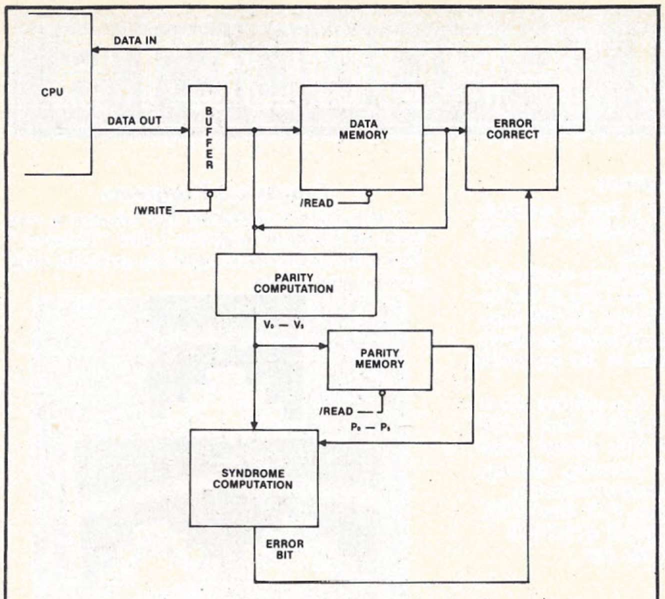


Figure 6. Bus Control.

CONCLUSION

Implementation of an error correcting memory requires the addition of six bits of memory and a few chips of random logic. The payoff is in terms of increased reliability. A system with error correcting memory will require fewer service calls and if a service call is generated for other reasons, the memory can be brought back to full operation.

As service costs rise and are measured in hundreds of dollars and memory costs decline, the user of error correction will increase. To make error correcting more practical, the semiconductor manufacturers are developing new circuits to perform the random logic functions.☐

NEW PRODUCTS

50/80 Interface

Mediamix has introduced a line of products related to interfacing the TRS-80 to the IBM Model 50 Electronic Typewriter.

These products include a hardware interface for connecting the Model 50 to the CPU or the Expansion Interface. The Mediamix 50/80 Interface plugs into the IBM Circuit board, so there's no soldering or modifications to the typewriter required.

Included with the 50/80 is a software driver program that lets the user LPRINT and LLIST plus access all of the IBM 50's special coded functions, like underlining, indenting, tabbing, numerical column justification and centering.

For details contact Mediamix, P.O. Box 8775, Universal City, CA 91608, (213) 475-9949.

CIRCLE INQUIRY NO. 121

Small Business Computer Systems

A line of small business computer systems has been introduced by Irvine, California based Alpha Micro. The AM series 1011, 1031 and 1051 computers are muti-tasking, multi-user, multiprocessor, time-sharing computers; models span the range from modest floppy disk systems to systems utilizing large capacity hard disk drives. The systems are designed for large scale programming in BASIC and other high level languages.

The AM series of computers offers many advantages commonly found on mainframe systems, at a fraction of the cost. The multi-user, multi-processor capability permits many users on the system at the same time.

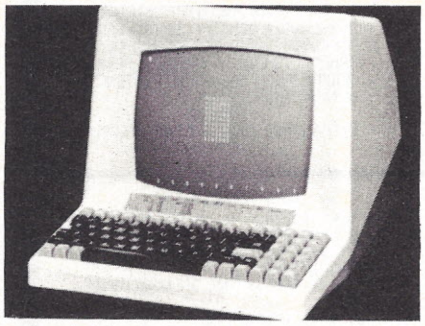
Each system includes a 16-bit processor, with two on-board serial I/O ports and one parallel I/O port. 64K of dynamic RAM memory is standard with each system.

For details contact Alpha Micro, 17881 Sky Park North, Irvine, CA 92714, (714) 957-1404.

CIRCLE INQUIRY NO. 122

miniMAS 2

Micro Application Systems announces miniMAS 2, Z8-based, high-performance, low-cost CRT family designed for large volume applications. The miniMAS 2 weighs approximately 20 pounds, measures 14W x 12H x 18D, and utilizes an external power supply requiring less than 50 watts.



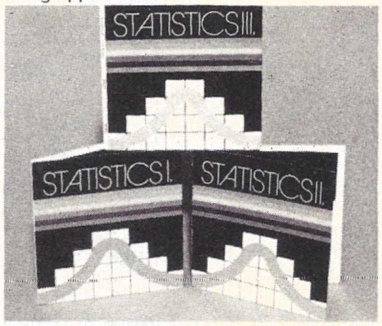
Standard hardware features include: 12" CRT, 7x9 dot matrix in a 9x13 field displaying all 128 ASCII codes, 24 lines of 39 or 80 characters, 25th line status display, 2 or 4K of memory for 1 or 2-page display, inverse or normal background, 16 baud rates for each of I/O and auxiliary ports and any combination of inverse, half intensity, blink, doublewide, underscore, and non-display attributes. Numeric pad, cursor and editing function keys, and reset key to terminate undesired action are standard.

For details contact Micro Application Systems, Inc., 5575 N. County Rd. 18, Minneapolis, MN 55442, (612) 559-0320.

CIRCLE INQUIRY NO. 123

Statistical Programs

Compucolor Corporation has released a new series of statistical programs called "Statistics." This three-disk series is especially useful for engineering applications.



These diskettes are entitled: Statistics I, Statistics II and Statistics III. Each disk contains five separate programs stored on a soft disk and comes complete with documentation.

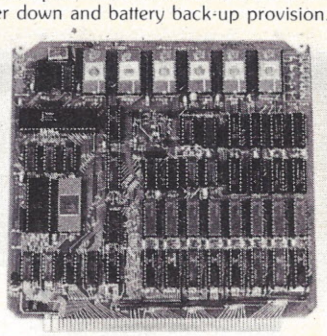
Common to all three packages is a file manager program that generates, maintains and displays files for use by other programs.

For details contact Compucolor Corp., P.O. Box 569, Norcross, GA 30091.

CIRCLE INQUIRY NO. 124

Process Control Card

The iCON/CPC-85 Process Control Card and Family, based on the popular 8085 microprocessor, offers power, performance and lots of I/O. It is supported with 12K or 24K of EPROM/PROM, up for 4K NMOS or CMOS RAM with power down and battery back-up provision.



Also included are three 16-bit timer/counters, two serial RS-232C ports, and 64 memory-mapped I/O lines: all on an 8 x 7.5 in. card. The CPC-85 is fully supported with expansion I/O cards, card cage, software and much more.

For details contact Process Control Inc., 2211 S. 48th St., Suite C, Tempe, AZ 85282, (602) 894-9105, Harry Latterman, Mktg. Manager.

CIRCLE INQUIRY NO. 125

Word Processing System

Spellbinder by California Pacific Computer Company includes all the features found on competitive high-quality word processing software — automatic word wrap, print formatting, proportional spacing, screen editing, justification block text manipulation, insertion, deletion, typeovers, search and replace, and emphasis/special character treatment.

Other features include full mailing list/mailing lable plus sort/merge and text/merge capacity, and legal numbering.

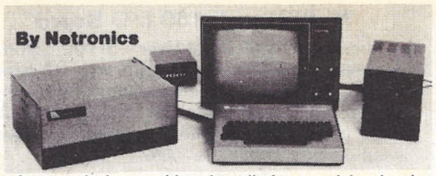
Spellbinder is available in IBM 8" single density, North Star double density, Micropolis quad density, Heath WH89, and Cromemco formats and is written in 8080 assembly language.

For details contact California Pacific Computer Co., 2601 Blackburn, Davis, CA 95616.

CIRCLE INQUIRY NO. 127

Start Computing For Just $129.95 With An 8085-Based Professional Computer Kit—

Explorer/85



By Netronics

100% compatible with all 8080A and 8085 software & development tools!

No matter what your future computing plans may be, Level "A"—at $129.95—is your starting point.

Starting at just $129.95 for a Level "A" operating system, you can now build the exact computer you want. Explorer/85 can be your beginner's system, OEM controller, or IBM-formatted 8" disk small business system...yet you're never forced to spend a penny for a component or feature you don't want and you can expand in small, affordable steps!

Now, for just $129.95, you can own the first level of a fully expandable computer with professional capabilities—a computer which features the advanced Intel 8085 cpu, *thereby giving you immediate access to all software and development tools that exist for both the 8085 and its 8080A predecessor (they are 100% software compatible)*—a computer which features onboard S-100 bus expansion—plus instant conversion to mass storage disk memory with either 5-1/4" diskettes or standard IBM-formatted 8" disks.

For just $129.95 (plus the cost of a power supply, keyboard/terminal and RF modulator, if you don't have them already), Explorer/85 lets you begin *computing* on a significant level... *applying* the principles discussed in leading computer magazines...*developing* "state of the art" computer solutions for both the industrial and leisure environment.

Level "A" Specifications

Explorer/85's Level "A" system features the advanced Intel 8085 cpu, an 8355 ROM with 2k deluxe monitor/operating system, and an 8155 ROM-I/O—all on a single motherboard with room for RAM/ROM/PROM/EPROM and S-100 expansion, *plus* generous prototyping space.

(Level "A" makes a perfect OEM controller for industrial applications and is available in a special Hex Version which can be programmed using the Netronics Hex Keypad/Display.)

Level "A" at $129.95 is a complete operating system, perfect for beginners, hobbiests, or industrial controller use.

PC Board: glass epoxy, plated through holes with solder mask • **I/O:** provisions for 25-pin (DB25) connector for terminal serial I/O, which can also support a paper tape reader ...provision for 24-pin DIP socket for hex keyboard/display...cassette tape recorder input...cassette tape recorder output...cassette tape control output...speaker output... LED output indicator on SOD (serial output) line...printer interface (less drivers)...total of four 8-bit plus one 6-bit I/O ports • **Crystal Frequency:** 6.144 MHz • **Control Switches:** reset and user (RST 7.5) interrupt...additional provisions for RST 5.5, 6.5 and TRAP interrupts onboard • **Counter/Timer:** programmable, 14-bit binary • **System RAM:** 256 bytes located at F8ØØ, ideal for smaller systems and for use as an isolated stack area in expanded systems...RAM expandable to 64k via S-100 bus or 4K on motherboard.

System Monitor (Terminal Version): 2k bytes of deluxe system monitor ROM located at FØØØ leaving ØØØØ free for user RAM/ROM. Features include tape load with labeling ...tape dump with labeling...examine/change contents of memory ...insert data...warm start...examine and change all registers...single step with register display at each break point, a debugging/training feature...go to execution address... move blocks of memory from one location to another...fill blocks of memory with a constant...display blocks of memory ...automatic baud rate selection...variable display line length control (1-255 characters/line)...channelized I/O monitor routine with 8-bit parallel output for high speed printer... serial console in and console out channel so that monitor can communicate with I/O ports.

System Monitor (Hex Version): Tape load with labeling... tape dump with labeling...examine/change contents of memory...insert data...warm start...examine and change all registers...single step with register display at each break point ...go to execution address. Level "A" in the *Hex Version* makes a perfect controller for industrial applications and can be programmed using the Netronics Hex Keypad/Display.

Hex Keypad/Display.

Hex Keypad/Display Specifications

Calculator type keypad with 24 system defined and 16 user defined keys. 6 digit calculator type display which displays full address plus data as well as register and status information.

Level "B" Specifications

Level "B" provides the S-100 signals plus buffers/drivers to support up to six S-100 bus boards and includes: address decoding for onboard 4k RAM expansion select-able in 4k blocks...address decoding for onboard 8k EPROM expansion selectable in 8k blocks...address and data bus drivers for onboard expansion...wait state generator (jumper selectable), to allow the use of slower memories...two separate 5 volt regulators.

Explorer/85 with Level "C" card cage.

Level "C" Specifications

Level "C" expands Explorer's motherboard with a card cage, allowing you to plug up to six S-100 cards directly into the motherboard. Both cage and cards are neatly contained inside Explorer's deluxe steel cabinet. Level "C" includes a sheet metal superstructure, a 5-card gold plated S-100 extension PC board which plugs into the motherboard. Just add required number of S-100 connectors

Level "D" Specifications

Level "D" provides 4k or RAM, power supply regulation, filtering decoupling components and sockets to expand your Explorer/85 memory to 4k (plus the original 256 bytes located in the 8155A). The static RAM can be located anywhere from ØØØØ to EFFF in 4k blocks.

Level "E" Specifications

Level "E" adds sockets for 8k of EPROM to use the popular Intel 2716 or the TI 2516. It includes all sockets, power supply regulator, heat sink, filtering and decoupling components. Sockets may also be used for soon to be available RAM IC's (allowing for up to 12k of onboard RAM).

Order A Coordinated Explorer/85 Applications Pak!

Experimenter's Pak (SAVE $12.50)—Buy Level "A" and Hex Keypad/Display for **$199.90** and get FREE Intel 8085 user's manual plus FREE postage & handling!

Student Pak (SAVE $24.45)—Buy Level "A," ASCII Keyboard/Computer Terminal, and Power Supply for **$319.85** and get FREE RF Modulator plus FREE Intel 8085 user's manual plus FREE postage & handling!

Engineering Pak (SAVE $41.00)—Buy Levels "A," "B," "C," "D," and "E" with Power Supply, ASCII Keyboard/Computer Terminal, and six S-100 Bus Connectors for **$514.75** and get 10 FREE computer grade cassette tapes plus FREE 8085 user's manual plus FREE postage & handling!

Business Pak (SAVE $89.95)—Buy Explorer/85 Levels "A," "B," and "C" (with cabinet), Power Supply, ASCII Keyboard/Computer Terminal (with cabinet), 16k RAM, 12" Video Monitor, North Star 5-1/4" Disk Drive (includes North Star BASIC) with power supply and cabinet, all for just **$1599.40** and get 10 FREE 5-1/4" minidiskettes ($49.95 value) plus FREE 8085 user's manual plus FREE postage & handling!

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

Netronics R&D Ltd., Dept. RE 10
333 Litchfield Road, New Milford, CT 06676

Please send the items checked below—

☐ **Explorer/85 Level "A" Kit** (ASCII Version), **$129.95** plus $3 p&h.

☐ **Explorer/85 Level "A" Kit** (Hex Version), **$129.95** plus $3 p&h.

☐ **8k Microsoft BASIC** on cassette tape, **$64.95** postpaid.

☐ **8k Microsoft BASIC in ROM Kit** (requires Levels "B," "D," and "E"), **$99.95** plus $2 p&h.

☐ **Level "B" (S-100) Kit, $49.95** plus $2 p&h.

☐ **Level "C" (S-100 6-card expander) Kit, $39.95** plus $2 p&h.

☐ **Level "D" (4k RAM) Kit, $69.95** plus $2 p&h.

☐ **Level "E" (EPROM/ROM) Kit, $5.95** plus 50¢ p&h.

☐ **Deluxe Steel Cabinet** for Explorer/85, **$49.95** plus $3 p&h.

☐ **ASCII Keyboard/Computer Terminal Kit** (features a full 128 character set, upper & lower case, full cursor control, 75 ohm video output convertible to baudot output, selectable baud rate, RS232-C or 20 ma. I/O, 32 or 64 character by 16 line formats, and can be used with either a CRT monitor or a TV set (if you have an RF modulator), **$149.95** plus $2.50 p&h.

☐ **Hex Keypad/Display Kit, $69.95** plus $2 p&h.

☐ **Deluxe Steel Cabinet** for ASCII Keyboard/Terminal, **$19.95** plus $2.50 p&h.

☐ **Power Supply Kit** (±8V @ 5 amps) in deluxe steel cabinet, **$39.95** plus $2 p&h.

☐ **Gold Plated S-100 Bus Connectors, $4.85** each, postpaid.

☐ **RF Modulator Kit** (allows you to use your TV set as a monitor), **$8.95** postpaid.

☐ **16k RAM Kit** (S-100 Board expands to 64k), **$199.95** plus $2 p&h.

☐ **32k RAM Kit, $329.95** plus $2 p&h.

☐ **48K RAM Kit, $459.95** plus $2 p&h.

☐ **64k RAM Kit, $589.95** plus $2 p&h.

☐ **16k RAM Expansion Kit** (to expand any of the above up to 64k), **$139.95** plus $2 p&h each.

☐ **Intel 8085 cpu User's Manual, $7.50** postpaid.

☐ **Special Computer Grade Cassette Tapes, $1.90** each or **3 for $5,** postpaid.

☐ **12" Video Monitor** (10 MHz bandwidth), **$139.95** plus $5 p&h.

☐ **North Star Double Density Floppy Disk Kit** (One Drive) for Explorer/85 (includes 3 drive S-100 controller, DOS, and extended BASIC with personalized disk operating system—just plug it in and you're up and running!), **$699.95** plus $5 p&h.

☐ **Power Supply Kit** for North Star Disk Drive, **$39.95** plus $2 p&h.

☐ **Deluxe Case** for North Star Disk Drive, **$39.95** plus $2 p&h.

☐ **Experimenter's Pak** (see above), **$199.90** postpaid.

☐ **Student Pak** (see above), **$319.85** postpaid.

☐ **Engineering Pak** (see above), **$514.75** postpaid.

☐ **Business Pak** (see above), **$1599.40** postpaid.

Total Enclosed $ ________
(Conn. res. add sales tax) By—
☐ Personal Check ☐ M.O./Cashier's Check ☐ Visa ☐ Master Charge
(Bank # _ _ _ _)

Acct. # ________

Signature ________ Exp. Date ____

Print Name ________

Address ________

City ________

State ________ Zip ________

☐ Send Me Information

By Netronics

ASCII/BAUDOT, STAND ALONE

Computer Terminal

COMPLETE FOR ONLY... **$149.95**

The Netronics ASCII/BAUDOT Computer Terminal Kit is a microprocessor-controlled, stand alone keyboard/terminal requring no computer memory or software. It allows the use of either a 64 or 32 character by 16 line professional display format with selectable baud rate, RS232-C or 20 ma. output, full cursor control and 75 ohm composite video output.

The keyboard follows the standard typewriter configuration and generates the entire 128 character ASCII upper/lower case set with 96 printable characters. Features include onboard regulators, selectable parity, shift lock key, alpha lock jumper, a drive capability of one TTY load, and the ability to mate directly with almost any computer, including the new Explorer/85 and ELF products by Netronics.

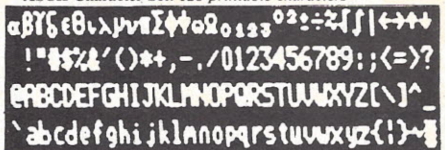
The Computer Terminal requires no I/O mapping and includes 1k of memory, character generator, 2 key rollover, processor controlled cursor control, parallel ASCII/BAUDOT to serial conversion and serial to video processing—fully crystal controlled for superb accuracy. PC boards are the highest quality glass epoxy for the ultimate in reliability and long life.

VIDEO DISPLAY SPECIFICATIONS

The heart of the Netronics Computer Terminal is the microprocessor-controlled Netronics Video Display Board (VID) which allows the terminal to utilize either a parallel ASCII or BAUDOT signal source. The VID converts the parallel data to serial data which is then formatted to either RS232-C or 20 ma. current loop output, which can be connected to the serial I/O on your computer or other interface, i.e., Modem.

When connected to a computer, the computer must echo the character received. This data is received by the VID which processes the information, converting to data to video suitable to be displayed on a TV set (using an RF modulator) or on a video monitor. The VID generates the cursor, horizontal and vertical sync pulses and performs the housekeeping relative to which character and where it is to be displayed on the screen.

Video Output: *1.5 P/P into 75 ohm (EIA RS-170)* • **Baud Rate:** *110 and 300 ASCII* • **Outputs:** *RS232-C or 20 ma. current loop* • **ASCII Character Set:** *128 printable characters—*



BAUDOT Character Set: *A B C D E F G H I J K L M N O P Q R S T U V W X Y Z - ? : * 3 $ # ( ) . , 9 0 1 4 ! 5 7 ; 2 / 6 8* • **Cursor Modes:** *Home, Backspace, Horizontal Tab, Line Feed, Vertical Tab, Carriage Return. Two special cursor sequences are provided for absolute and relative X-Y cursor addressing* • **Cursor Control:** *Erase, End of Line, Erase of Screen, Form Feed, Delete* • **Monitor Operation:** *50 or 60Hz (jumper selectable.*

Continental U.S.A. Credit Card Buyers Outside Connecticut

CALL TOLL FREE 800-243-7428

To Order From Connecticut Or For Technical Assistance, Etc. **Call (203) 354-9375**

Netronics R&D Ltd., Dept. PE-9
333 Litchfield Road, New Milford, CT 06776

Please send the items checked below—

☐ **Netronics Stand Alone ASCII Keyboard/Computer Terminal Kit, $149.95** plus $3.00 postage & handling.

☐ **Deluxe Steel Cabinet** for Netronics Keyboard/Terminal In Blue/Black Finish, **$19.95** plus $2.50 postage and handling.

☐ **Video Display Board Kit** alone (less keyboard), **$89.95** plus $3 postage & handling.

☐ **12" Video Monitor** (10 MHz bandwidth) fully assembled and tested, **$139.95** plus $5 postage and handling.

☐ **RF Modulator Kit** (to use your TV set for a monitor), **$8.95** postpaid.

☐ **5 amp Power Supply Kit** In Deluxe Steel Cabinet (±8VDC @ 5 amps, plus 6-8 VAC), **$39.95** plus $2 postage & handling.

Total Enclosed (Conn. res. add sales tax) $________
By—
☐ Personal Check ☐ Cashiers Check/Money Order
☐ **Visa** ☐ **Master Charge** (Bank # _ _ _ _)

Acct. # ________

Signature ________ Exp. Date ________

Print Name ________

Address ________

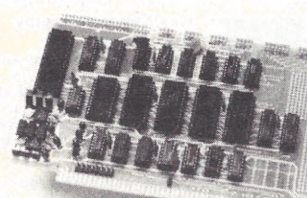
City ________

State ________ Zip ________

☐ Send Me More Information

Multi-User S-100 I/O Board

Micromation's Multi-User S-100 Board features four RS232 serial ports with full handshaking capability, three programmable timers, two bus-driving parallel output ports, three parallel input ports with handshake capability, plus wire wrap area for custom circuitry.



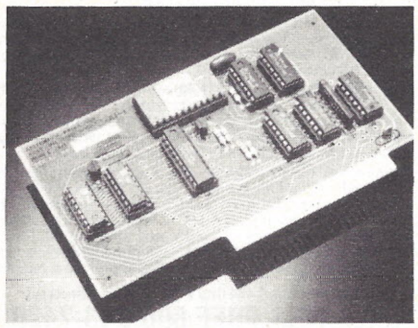
The four serial RS-232 I/O ports employ individual switch-selectable baud rates to 9600 baud. The four ports employ 8551 UARTs to deliver full handshaking and full interrupt support. All UARTs and timers can be interrupt-driven for fast system operation.

For details contact Micromation, 488 Cowper St., Palo Alto, CA 94301, (415) 328-5181, David Carlick.

CIRCLE INQUIRY NO. 126

Arithmetic Processor Unit

The Model 7811B by California Computer Systems is designed to increase the execution speed of Applesoft II programs as well as to increase the number of math functions available to the programmer.



The card employs the AMD9511 APU. It is a hardware floating point unit powerful enough to decrease program execution time by up to one order of magnitude.

For details contact California Computer Systems, 250 Caribbean Dr., Sunnyvale, CA 94086.

CIRCLE INQUIRY NO. 128

Thermal Printer for Apple

Silentype™ is an advanced text and graphics printer for the Apple II. It is a quiet, low-cost peripheral that will enable Apple II users to print on paper copies of anything that the computer can display on a video monitor or television screen.

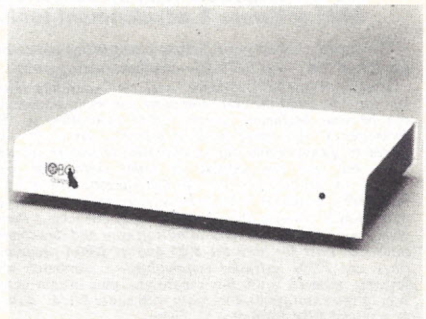
Instead of a conventional character-printing hammer mechanism, the printer uses a 7-dot thick-film printing element to produce 5x7 dot characters and graphics on standard thermal paper. It prints up to 80 characters per line on 8½ inch wide roll-fed paper.

For more information contact Apple Computer, Inc., 10260 Bandley Dr., Cupertino, CA 95051, (408) 996-1010, Jean Richardson.

CIRCLE INQUIRY NO. 129

Expansion Interface for TRS-80

Lobo Drives International announced the addition of an expansion interface for the Radio Shack TRS-80 computer. The Model LX80 enhances system performance by expanding memory storage capacity up to 40 million bytes.



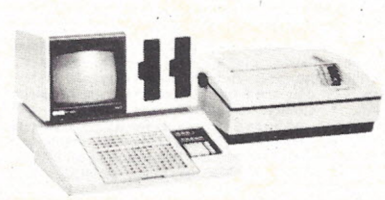
It provides facilities for up to 32K of RAM and offers a second serial port. A switch permits overriding the keyboard ROM for booting in diagnostics and customized operating systems.

For details contact Lobo Drives Int'l., 935 Camino Del Sur, Goleta, CA 93017, Mike Mock.

CIRCLE INQUIRY NO. 130

Small Business Computer

The BC-5000 from Panasonic is a desk-top computer which features a one-touch keyboard and double-sided, double density floppy disk drives. The unit can be utilized as a small business computer or intelligent terminal for distributed data processing.



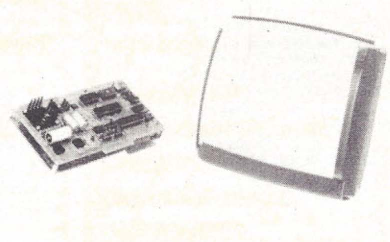
The unit is well-suited for order entry, inventory control, billing and a variety of applications which can benefit from the increased productivity obtained when using the one-touch keyboard.

For more information contact Panasonic, One Panasonic Way, Secaucus, NJ 07094.

CIRCLE INQUIRY NO. 131

Touch Screen Digitizer

The Touch Screen Digitizer from TSD Display Products is designed for use with 12-inch diagonal CRTs. The Touch Screen provides one solution for interfacing personnel who have no



computer and data processing experience with a database that is stored in a computer. It eliminates the need for keyboards and light pens. The operator's attention is constantly focused on the screen.

For details contact TSD Display Products, Inc., 35 Orville Dr., Bohemia, NY 11716.

CIRCLE INQUIRY NO. 132

Disk Sort/Merge System

A disk sort/merge system "DSM" is available for both the TRS Mod-I and Mod-II. DSM is a self-contained system written in machine language ready for immediate use.

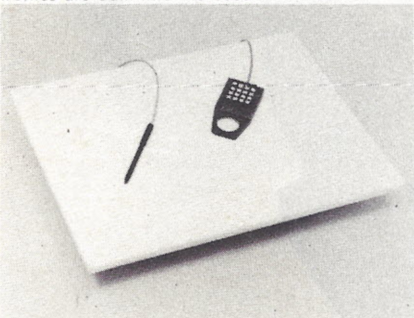
DSM sorts large multiple diskette files on a minimum one drive Mod-II or two drive Mod-I disk system; physically rearranges all records; sorts random files created by BASIC, including sub-records spanning sectors; sorts on one or more fields in ascending or descending order. It also provides optional output field deletion, rearrangement, and padding.

DSM is available from Racet Computes, 702 Palmdale, Orange, CA 92665, (714) 637-5016.

CIRCLE INQUIRY NO. 133

Industrial X-Y Digitizer

The Digi-Pad single-unit digitizer tablet measures 1.7 inches high, has no adjustments, requires no preventive maintenance and all electronics are built into the base of the tablet.



Designed for applications requiring the conversion of graphic data into digital form, Digi-Pad has applications in computer aided design, entry of menu data, analysis of statistical data and more.

Contact GTCO Corp., 1055 First St., Rockville, MD 20850, (301) 279-9550.

CIRCLE INQUIRY NO. 134

Instant Processor Switcher

Dynatech Data Systems has available a front end processor (FEP) switching system that provides both network cost savings and increased reliability. This system allows a single spare processor to serve as a backup for multiple on-line processors, and has the ability to switch all channels from a failed processor to the spare processor.

The system consists of a series of multi-channel A/B fallback switches with the spare FEP chained to the B positions on each switcher. A remote control panel with interlock circuitry provides instant switching while preventing more than one set of communications lines from being connected to the spare FEP simultaneously.

Patching access to every channel permits individual channel reconfiguration and non-interrupting monitor/signal breakout. Operation is controlled by pushbuttons with a keyswitch for system security.

Contact Dynatech Data Systems, 7644 Dynatech Ct., Springfield, VA 22153, (301) 279-9550.

CIRCLE INQUIRY NO. 135

Microcomputer Printer Interface

The I/OMaster S-100 Interface Board from MicroPro allows flexible use of either lower cost letter-quality printers and/or high speed line printers within the same microcomputer configuration.

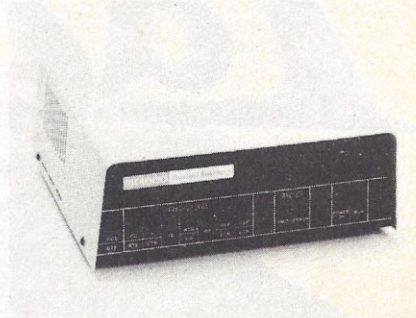
Combining four boards in one, I/OMaster features two each serial and parallel ports as well as an eight level interrupt control and dual interval timer circuitry. To insure that keystrokes and transmitted communication data are not lost during task switching operations, I/OMaster's two 8251-based serial ports each have built-in 32-character FIFO buffers. All I/OMaster options are DIP switch selectable.

For details contact MicroPro International Corp., 1299 Fourth St., San Rafael, CA 94901, (415) 457-8990.

CIRCLE INQUIRY NO. 136

Parallel/Serial Converter

The PSC/4000, offered by The Standard Register Company, is an easy-to-use, solid state microprocessor-based unit. Both interfaces are programmable. In addition, data can be edited, reformatted, etc. by the unit prior to output.



The PSC/4000 interfaces a variety of quantitative measuring devices—scales, counters, etc.—with mini and microcomputers, CRTs, Teletypewriters and serial printers.

For details contact The Standard Register Co., P.O. Box 1167, Dayton, OH 45401.

CIRCLE INQUIRY NO. 138

Treasure Hunt Software Game

CASTLE, a fantasy simulation game in which the player searches for treasure in a magical world, is available in North Star disk BASIC (version 6 or later). The game provides detailed descriptions of the locations, objects and situations encountered by the adventurer, and accepts English-like commands from a vocabulary of over 150 words.

CASTLE requires a system with 32K bytes of memory, and a single or double density North Star disk drive. For more information contact International Computing and Robotics, 4920 Harmony Way, San Jose, CA 95130.

CIRCLE INQUIRY NO. 137

The Business "Dream" Machine?

By David R. Fuller & Harold Henrich
Lazor Systems, Inc.
1050 E. Duane Avenue, Sunnyvale, CA 94086

Although the typical small business system is designed to serve a single user performing a single task, even the smallest business or district office outgrows single-minded capability. The system may have been geared to accounting, but someone wants word processing. Why not use it for inventory control, production scheduling, or management analysis? As the range of applications and the number of users proliferate, the system cannot handle the increased load.

Lazor Systems, Sunnyvale, CA, has developed a system that solves this problem in two ways. First, the basic system can perform multiple tasks concurrent with serving multiple users and stations. Secondly, the system, designed in a modular fashion, allows for extensive growth in capability at minimum cost.

The Lazor is an advanced multi-bus multi-processor small business system that collects, analyzes, decodes, executes and records multi-channel information at differing throughput rates. While one 16-bit processor serves as the master CPU, one or more 16-bit I/O processors can be added to handle multiple input/output operations simultaneously. Parallel execution by multiple processors is accomplished by task partitioning and resource allocation.

The Lazor can be used as a master CPU supporting up to 16 terminals or as a district office front-end processor communicating with a central IBM host via a communications controller.

Each processor in the network handles two or more of the five basic functions:

- Task allocation and resource management
- Information processing
- Information concentration and temporary storage
- Local input/output and hardware control
- Remote input/output and communications

Functional organization of the master modules, on the Lazor Advanced Multibus, each with its own high speed buffer memory, minimizes bus contention allowing effective

addressing and aggregate data transfer rate of two megabytes per second.

INPUT/OUTPUT PROCESSORS

The I/O processor, in conjunction with the I/O controller, is designed for optimum performance of I/O operations. The main processor sets up the I/O task in the main memory and notifies the I/O processor to start execution, then returns to application execution. The I/O processor initiates operation by directly addressing the requested device and giving it a command. The I/O processor controls the direct memory access (DMA) transfer to buffer memory, transfers data to and from main memory and devices, handles all input/output interrupts, and notifies the master processor when the requested I/O operation is complete.

The I/O processor controls devices with a wide range of speeds. Multiple devices can interleave transfers of data to and from memory utilizing the full band-width of the I/O processor with no degradation. In addition, the I/O processor performs error checking on all input/output operations, retries errors, informs the task requesting the I/O operation of any non-recoverable errors, and at user option, keeps a log of errors on a disk data set.

This frees the main processor for task management and execution of application programs, increasing total throughput.

INTERRUPTS

The system eliminates the need for high overhead polling techniques via three classes of interrupts:

- Class 1: Non-maskable
- Class 2: Supervisor call
- Class 3: I/O interrupts

Class 1 interrupts are non-maskable to immediately alert the system to error or exception conditions such as power fail warning, invalid address, storage protect violation, divide exception, double bit error and single cycle.

Class 2 supervisor call interrupts are programmed instructions under user control to call supervisor routines.

Class 3 I/O interrupts are software maskable and serviced by the I/O processor using a double indirect vectoring scheme. This sets a pointer to any of the unique device parameters and automatically branches to a common or unique service routine.

Programmable priority at the control and device levels permit the supervisor program to define or dynamically change the interrupt priority level of any device. For example, the main operator console is a priority level 1, but when another terminal needs real time processing, device priority can be changed to level 1, and the main console switched to a lower priority.

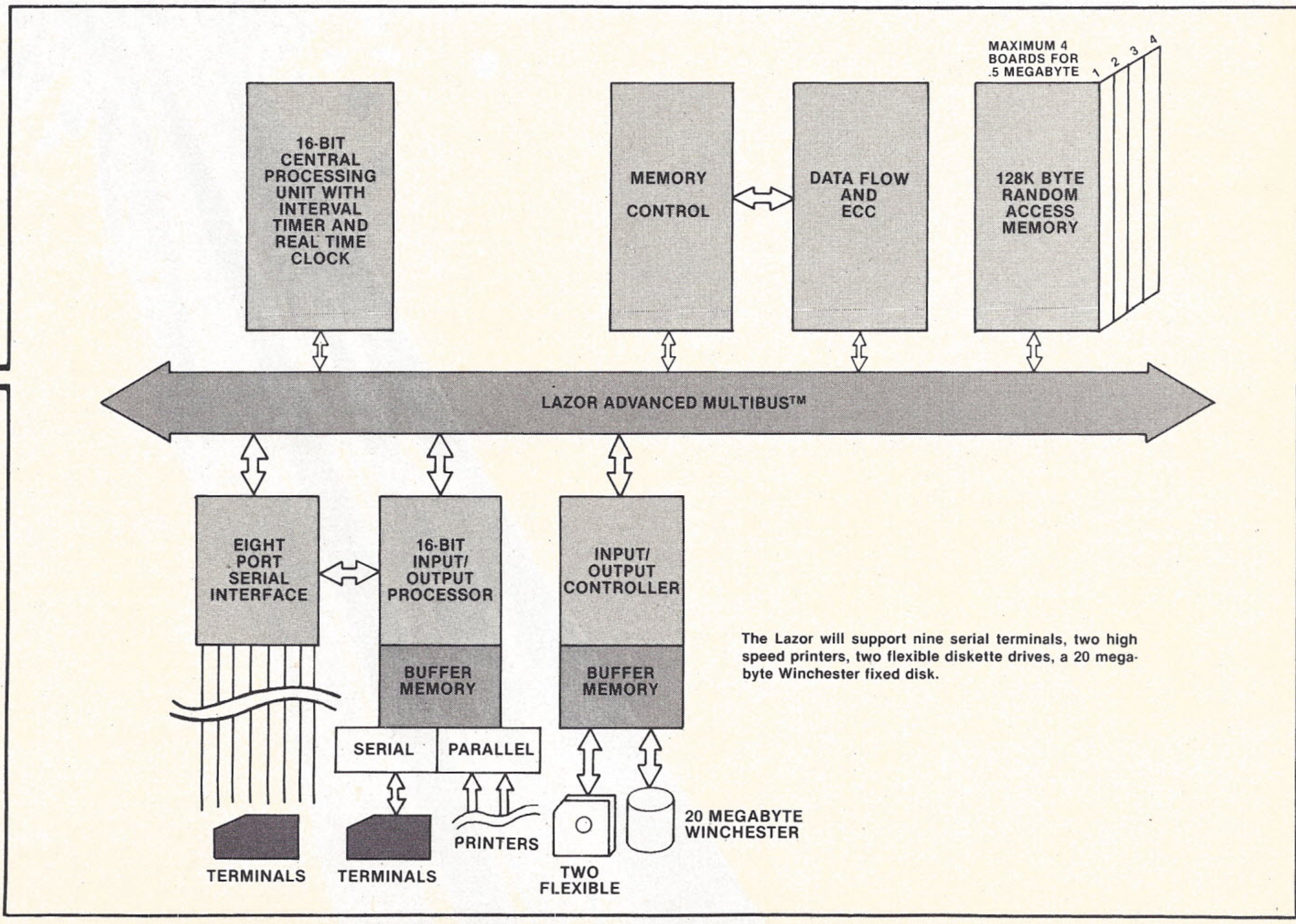
There are three ways to mask interrupts. Processor level masking masks all I/O interrupts. Control level masking masks I/O on particular priority levels, and device masking masks a particular device. This gives the Lazor excellent control of its available resources.

MLX OPERATING SYSTEM

Lazor MLX is an interrupt drive, multi-task operating system designed to operate with the system architecture to provide:

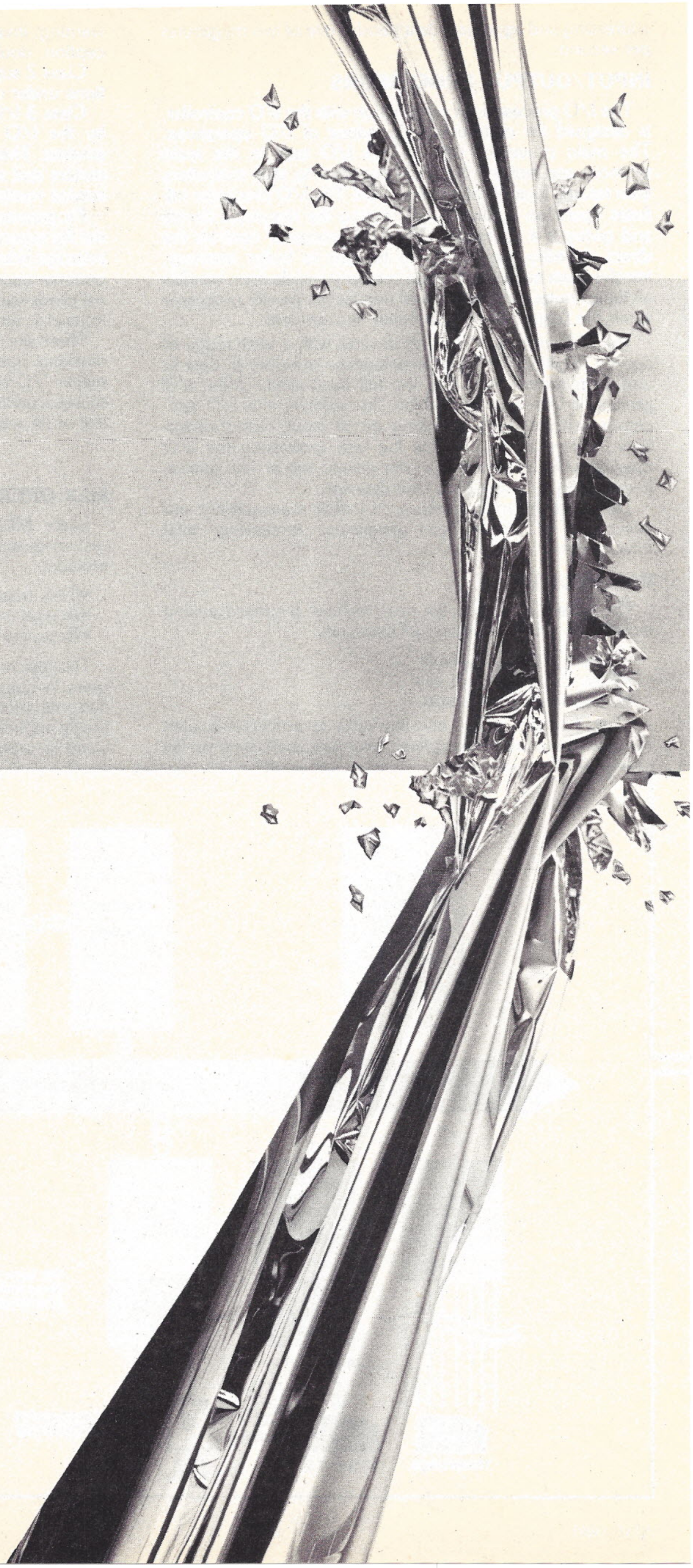
- Task management
- Access methods
- Resource management

The task manager schedules tasks for up to 16 concurrent users. Its unique run time monitor and priority scheme provides dynamic priority switching, at specified intervals, to guarantee timely application completion. Log-on services are also provided to prevent unauthorized access to the system.



The Lazor will support nine serial terminals, two high speed printers, two flexible diskette drives, a 20 megabyte Winchester fixed disk.

ALTOS
COMPUTER SYSTEMS



*Z80 is a registered trademark of Zilog, Inc.
**CP/M is a registered trademark of Digital Research, Inc.

ALTOS BREAKS THE MICRO BARRIER.

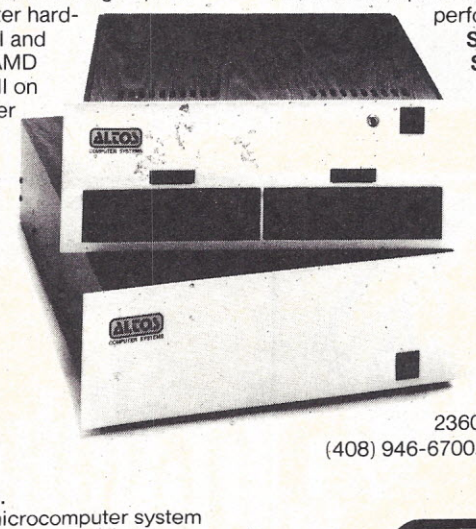
Yesterday, microcomputer meant micro performance. Once you outgrew it, you had to step up to a mini. Which meant a big step up in price.

Today, there's the new Altos ACS8000-6 single-board microcomputer system.

It's the first system for the OEM, small businessman and personal user, that offers minicomputer performance and minicomputer storage capacities—at a microcomputer price.

MULTI-USER, WINCHESTER STORAGE, FLOPPY BACK UP: $14,260.

The new Altos ACS8000-6 is a highly advanced Z80* based microcomputer system with high-speed RAM, floppy disk and Winchester hard-disk controllers, DMA, six serial and two parallel I/O ports and the AMD 9511 floating point processor all on a single board. A typical four-user system configuration with two megabytes of Shugart floppy and 29.0 megabytes of Shugart Winchester storage, including CPU and 208K bytes of RAM, costs only $14,260—compared to $30,000 or more for a similar minicomputer system. And that adds up to <u>mini performance at less than half the cost!</u>



MULTI-USER EXECUTIVE SUPPORTS FOUR INDEPENDENT USERS RUNNING CP/M COMPATIBLE PROGRAMS.**

This revolutionary new microcomputer system features the MP/M** Multi-User Executive software program that's unique in two ways. It includes a multi-user CP/M capability and the ability to handle Winchester-type hard disks. The advanced Z80 operating program supports four independent CP/M compatible programs in any of six popular languages: BASIC, FORTRAN, COBOL, PASCAL, APL, C, and a large assortment of additional business application packages. MP/M is compatible with both the 1.4 and 2.0 versions of Digital Research's CP/M, which means programs based on either version can run under MP/M without modification.

With MP/M at the helm, your Altos ACS8000-6 system can support up to four simultaneous users with 48K bytes of RAM each plus 58 megabytes of Winchester storage and 4 megabytes of floppy back up. And that adds up to the first microcomputer to give you the power and performance of a minicomputer.

SINGLE-USER, HARD-DISK SYSTEMS START AT $9450.

The Altos ACS8000-6 series. It's a barrier breaker in every sense. Our entry-level, single-user, hard-disc system with floppy back up is priced under $10,000 and even our 4-user CP/M model is available for under $12,000. And all configurations are easily upgraded. For specific details about pricing or performance, call or write: Altos Computer Systems, 2360 Bering Drive, San Jose, CA (408) 946-6700. TELEX 171562 ALTOS SNA

ALTOS
COMPUTER SYSTEMS

CIRCLE INQUIRY NO. 3

The MLX access methods provide the user with three high level disk access methods. The Indexed Sequential Access Method (ISAM) allows the user to access disk resident data by a key record identifier. Relative record access allows random access simply by specifying record number. Sequential access provides access to records in a sequential manner beginning with the first record in a file. In addition, password verification is provided to protect user data sets from unauthorized access.

The resource manager controls the system's resources by providing memory management, timer services, and input/output device allocation. Dynamic allocation and deallocation of memory, as well as memory partitioning by task, are provided by the memory management routines.

The resource manager's timer routines provide accurate time of day for time stamping applications and messages. It also provides the task manager with its requested time intervals for priority switching.

The input/output control routines assign physical I/O devices to logical I/O names allowing individual applications to be independent of particular I/O configurations.

SYSTEM CONFIGURATION

Two system series are available: the economical flexible disk-based E-series and the cartridge disk-based S-series.

The basic Lazor is a 2.4 megabyte flexible disk computer system with a 16-bit CPU, 64K bytes of dynamic RAM memory with ECC, one flexible diskette controller, and one Winchester disk controller, eight serial interface ports, two 1.2 megabyte flexible diskette drives, and a 16 position back panel allowing space for expansion.

When the user needs more capability, he can upgrade to a higher model that adds an I/O processor, 64K bytes of additional memory, a second flexible diskette controller (one for each drive), and two parallel interfaces. If the user needs still more capability, he can move to a unit that adds a 20 megabyte Winchester drive and a ninth serial interface. Upgrades are via plug-in additions.

The S-series begins with the entry-level 32 megabyte cartridge disk computer system. The basic system includes the 16-bit CPU, 128K bytes of dynamic RAM memory with ECC, a cartridge disk controller, eight serial interfaces, a 32 megabyte cartridge disk subsystem with 16 megabytes of removable storage and 16 megabytes of fixed storage, and a 20 position back panel for expansion.

One model adds an I/O processor, two parallel interfaces, a ninth serial interface, an additional 32 megabytes of fixed disk storage for a total of 64 megabytes. It adds still more capability by adding a second 16-bit CPU, another 128K memory, and an additonal 32K megabytes of fixed disk storage to bring the total storage capacity to 96 megabytes. Again, the upgrades are made without changing the basic system hardware or software.

Each model can be configured to meet specific application requirements. For example, additional I/O controllers provide for large terminal networks. Parallel interfaces provide the necessary data rates for complex graphics. Disk drives, removable or fixed, can be added in various increments to satisfy applications requiring more storage capacity.

The system will support memory expansion to one megabyte, up to four double density flexible diskette drives, up to 16 local or remote operator terminals, and multiple printers. With optional communications support, the system can handle bisync and async data communications protocols and includes a teletype adapter, providing voltage level compatibility with EIA RS232 interfaces.

As with other systems, the user can select from a variety of output devices. Lazor offers interfacing for letter quality daisywheel printers, matrix printers (operating under serial or parallel), high speed line printers and printer/plotters.

 CIRCLE INQUIRY NO. 76

PERIPHERALS

- Flexible disk drive (double density, double sided, 1.2 megabyte)
- 140 cps matrix printer (serial interface)
- 150 cps matrix printer (parallel interface)
- 300 lpm printer (parallel interface)
- 45 cps letter-quality printer
- CRT intelligent terminal
- Winchester disk (20 megabytes)
- Cartridge disk subsystem (32 megabytes, 16 megabytes, fixed; 16 megabytes, removable)
- Cartridge disk subsystem (64 megabytes, 48 megabytes, fixed; 16 megabytes, removable)
- Cartridge disk subsystem (96 megabytes, 70 megabytes, fixed; 16 megabytes, removable)

SOFTWARE

Lazor MLX multi-level executive operating system
Lazor BASIC compiler
Lazor COBOL compiler
Applications programs
General ledger
Accounts payable
Accounts receivable
Payroll with cost accounting
Order entry with inventory control
Text editor
Word processing

RELIABILITY

Modular design and high-level large-scale integration offer improved reliability through reduction in the number of components and interconnections. Wiring between printed circuit boards has been eliminated by plugging all boards into a common bus back panel. Logic handles worst case timing and environmental conditions.

A power-fail interrupt protects against failures. When input line voltage drops below required levels, the system detects the failure and sends a "power warning interrupt" to the main processor. The active data files are immediately written in disk storage.

MEMORY ERROR DETECTION

When an application accesses a memory address that exceeds memory capacity, in most conventional mini and micro-based systems, the user is not notified. New data is either written over existing data in other memory locations or simply lost. In Lazor, the memory notifies the main CPU and operator that a program has attempted to access a location outside of memory boundaries and was unsuccessful.

Memory-write protection is a standard feature in all systems. The Lazor operating system provides each user with a protected area of memory, especially valuable when shared by multiple users.

Error Check and Correction (ECC) minimizes system failures. ECC will detect all single, double, and some multiple bit errors and correct all single-bit errors.

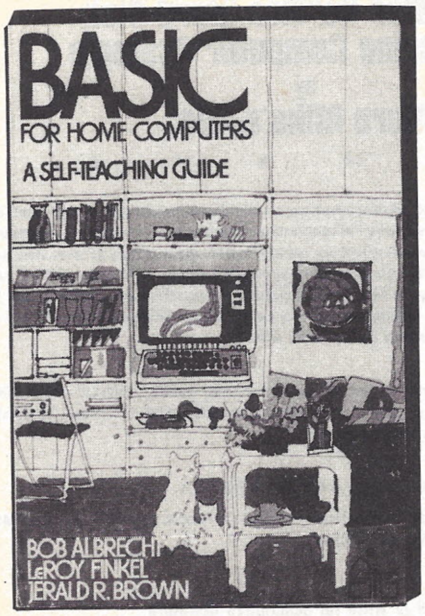
DIAGNOSTICS

Diagnostics are run each time the system is powered-on or system reset is depressed for initialization. Malfunctions are detected by the sequence-driven master test module and displayed on the system console. Modular board design simplifies isolation of system failures.

SECURITY

Lazor offers both software and hardware security keys. System reset and power-on are controlled by a key lock device.□

from Books in BASIC...



BASIC For the Home Computer
By Bob Albrecht, LeRoy Finkel, and Jerald R. Brown
Order No. 0-471-03204-2
336 pages $6.95

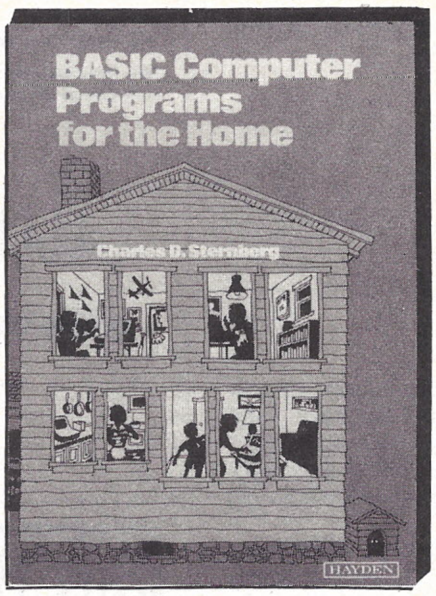
Learn how to read, write and understand BASIC, including applications and games for the hobbyiest or for anyone who wants to use small computers in home, school or office.

32 BASIC Programs for the TRS-80
By Tom Rugg and Phil Feldman
Order No. 0-918398-27-4
290 pages $15.95

Now that you've bought yourself a TRS-80 computer, what do you do with it? You will find the answer to that and many more questions in this book. Each of the 32 chapters fully documents a different program. The authors explain how to make simple modifications, and if you have a good working knowledge of BASIC, you can devise and implement your own program changes. Valuable to both the noive and "old hand".

Atari BASIC
By Bob Albrecht, Leroy Finkel, and Jerald R. Brown
Order No. 0-471-06496-3
$5.95

Learn to read, write, and understand BASIC-all in a self-intructional format that requires no special math or science background or previous experience with a computer. Supplies all necessary information, from how to get started to a list of popular computing magazines; contains numerous applications and games.



BASIC Computer Programs for the Home
By Charles D. Sternberg
Order No. 0-8104-5154-9
330 pages $8.95

A comprehensive book of practical home application programs that will be helpful to both the novice and the experienced owner in increasing the usefulness of all home computers. Each program is documented with a description of its functions and operation, a listing in BASIC, a symbol table, sample data, and one or more output samples.

Some Common BASIC Programs
By Lon Poole and Mary Borchers
Order No. 0-931988-06-3
220 pages $12.50

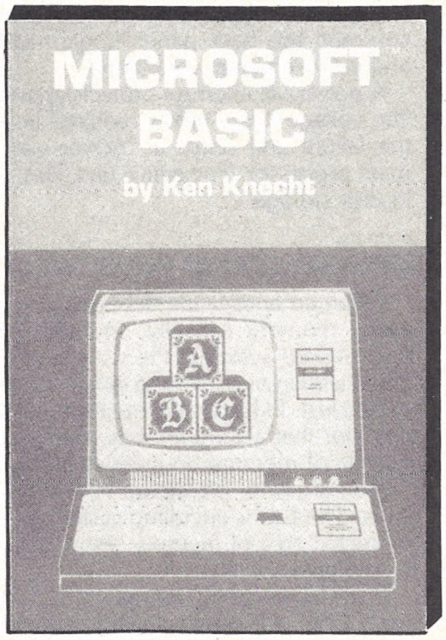
Program listings and documentation for 76 short programs covering financial, mathematical, statistical, and general interest applications.

Instant BASIC
By Jerald R. Brown
Order No. 0-918398-21-5
159 pages $9.95

For the microcomputer enthusiast or the user of DEC's BASIC plus language, here is an "active participation" workbook designed to use with your home computer. It's an easy, painless way to learn BASIC.

BASIC 2nd Edition
By Bob Albrecht, LeRoy Finkel and Jerald Brown
Order No. 0-471-03500-9
325 Pages $5.95

Revised and updated to make it simpler than ever to learn BASIC without the need for special math or science background. Shows how to use the computer as a tool in any field from education to the humanities.



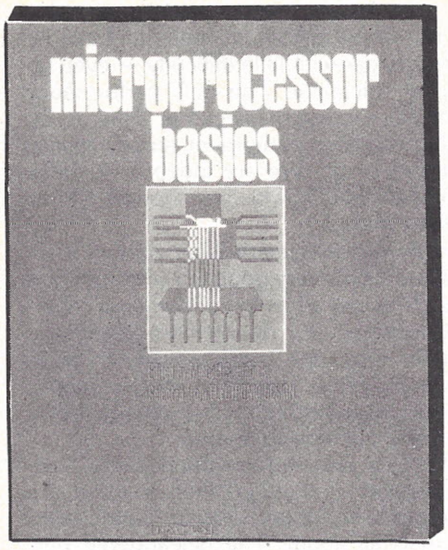
Microsoft BASIC
By Ken Knecht
Order No. 0-918398-23-1
225 pages $9.95

Illustrates concepts presented in such a way that commands valid at each succeeding level of BASIC can be easily used and understood. By starting with the simplest and most commonly used commands and then progressing on to more complex BASIC, the author illustrates how the more powerful versions of the language can save valuable programming time and effort.

to Basic Books

DATA DYNAMICS TECHNOLOGY

has them all!



Microprocessor BASICS
Edited By Michael Elphick
Order No. 0-8104-5763-6
224 pages $11.95

Aimed at the design engineer, seperate sections are devoted to each of the popular microprocessors available. Each section also discusses one available model, explaining its advantages, disadvantages, architecture, and capabilities.

Computer Dictionary
By Donald Spencer
Order No. 0-89218-038-2
160 pages $5.95

This current and comprehensive dictionary contains about 2500 words, phrases, and acronyms used in connection with computers. The keynote of this book is clarity- without sacrifice of authority and definitional precision. All definitions are simple, and stand as independent units of explanation. This book was written for all those who, in their work or in their studies, become involved with computer technology.

The Semiconductor Memory Book
By Intel Marketing Communications
Order No. 0-471-03567-X
524 pages $14.95

A complete and extensive collection of semiconductor memory application and specification information complied by one of the world's largest suppliers of memories and microprocessors.

Microprocessor Interfacing Techniques
By Austin Lesea, Rodnay Zaks
Order No. 0-89588-029-6
464 pages $15.95

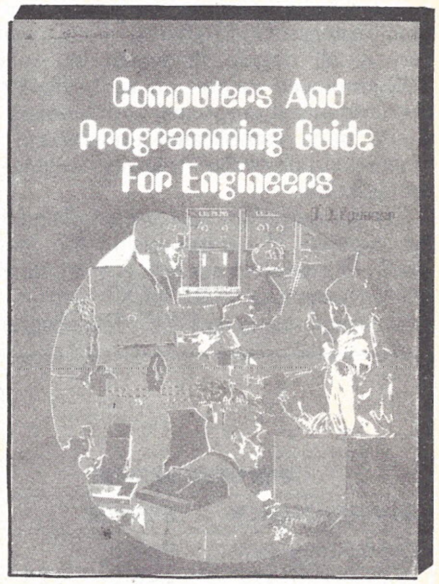
Microprocessor interfacing is no longer an art. It is a set of techniques, and in some cases, just a set of components. This book introduces basic interfacing concepts, and presents in detail, implementation techniques for both hardware and software. It covers the essential peripherals, from keyboard to floppy disk, as well as standard buses and introduces basic troubleshooting techniques.

Microcomputer Primer
By Mitchell Waite and Michael Pardee
Order No. 0-672-21653-1
368 pages $11.95

Build your knowledge and appreciation of microcomputers from the ground up with this revised second edition on those enormously popular typewriter-size computers. Covers the most recent advances in microcomputer technology and contains an introduction to microcomputers, basic computer concepts, electronics of microcomputer functions and many more.

Computers & Programming Guide For Engineers
By Donald D. Spencer
Order No. 0-672-20950-0
288 pages $12.95

Written for the engineer or scientist, this book explains what a computer is, how it works, and how it can be used to the best advantage. It discusses various types of digital computers, input/output devices, storage devices, storage media, and also covers the steps the engineer uses in problem solving on a computer. Includes several problem-oriented languages and languages for simulations.



DATA DYNAMICS TECHNOLOGY P.O. Box 1234, Cerritos, CA 90701 **6/80**

Name (Print) ____________________
Address ____________________
City ____________________ State ________ Zip ________

Please send me:

Description	Qty	Price	Total	Book Order #	Qty	Price	Total

Shipping & Handling Charges
Books $ .75 ea. U.S., $1.50 ea. Foreign

TOTAL ORDER $ ________
***TAX $** ________
SHIPPING & HANDLING $ ________
TOTAL ENCLOSED $ ________

☐ master charge ☐ American Express ☐ VISA # ____________ ☐ Check or M.O.(U.S. Funds drawn on U.S. bank)

Exp. Date ________ Signature ____________________

For European delivery contact **INTERFACE AGE Europe,** Dahlienstr. 4, D-8011 Munchen-Vaterstetten, West Germany.
*California residents add 6% sales tax. Availability and prices quoted subject to change without notice.
Please allow six weeks for delivery. You may photocopy this page if you wish to keep your INTERFACE AGE intact.
Orders *cannot* be shipped unless accompanied by payment, including shipping & handling and tax where applicable.
DATA DYNAMICS TECHNOLOGY, A Division of INTERFACE AGE Magazine (213) 926-9548

Assignment: BENCHMARK

Try this for an afternoon's entertainment: Walk into a computer factory, whip out a simple-looking 15-line BASIC program, and ask to run it on its most cherished product. While keying it in, mention that you intend to publish the results in a national magazine, along with the best performances of a dozen or so of their most honored competitors. If that doesn't immediately result in rapid escort to the nearest exit, you will find yourself in the center of plenty of attention from more apprehensive-looking executives than those assembled for the last IRS audit.

We have taken part in that scenario quite a few times in the past several months, and have learned a lot about benchmarking as an art. In its simplest form, a benchmark is a task that can be run on different computer systems to compare the performance of each. Ideally, the task should be one that is meaningful to the intended application of the computer in your business. Unfortunately, that's usually possible only to a very small degree. It would be enlightening to run a 100-employee payroll on two different computer systems (using identical programs) and compare the results; but that would be a complicated undertaking and would involve neutralizing all of the irrelevant variables that might obscure the outcome.

For our purposes, we have simplified the test, admittedly at the expense of compromising the meaningfulness of the result. Our benchmark task is a simple BASIC program that utilizes two nested FOR/NEXT loops to "discover" all of the prime numbers up to 1000. The program is so simple it can easily be adapted to run on all BASIC variants, and can be translated into other languages with little difficulty — even on a programmable pocket calculator. The only performance criteria we are interested in is speed of execution. This can be measured without using precision equipment, since a typical run will take ten minutes or so. We are indebted to Mike Simmons, inventor of the HEX29, who first showed it to us during a visit to his laboratory.

Some of the computer factory experts have told us it's a dumb program that doesn't do a very good job of computing prime numbers. It exercises but a fraction of the large repertory of capabilities featured by any decent computer, so it tends to favor the simple languages at the expense of those that otherwise might be far more capable in other respects. And finally, spitting out prime numbers is a scant test of a computer as a meaningful addition to society.

All of that is true, of course, but we have developed a fondness for our adopted program, and will continue to use it as a test of one small aspect of a computer's worth. Speed, after all, is what computing is all about. The beleaguered purchaser has precious little help in quantifying the differences amongst the current cornucopia of offerings. While far from being the ultimate test of processing speed, we think that printing "997" (the last prime less than 1000) in under four minutes says more about a system's capability than quoting the clock rate of the fastest quartz crystal in the machine.

Table 1 shows the results we have gathered so far — from the giant time-sharing engine of a large university's com-

Table 1. Benchmark Performance Data

COMPUTER SYSTEM		PROCESSOR			SOFTWARE		BENCHMARK
MANUFACTURER	SYSTEM	TYPE	BITS	SPEED	OPER SYS	LANGUAGE	RUN TIME
Digital Equipment	PDP-10	n/a	36	n/a	TOPS-10	BASIC	65 sec
Digital Microsystems	HEX29	2900	16	6 MHz	HOST	HBASIC+	143 sec
Alpha Micro	AM-100/T	WWD16	16	3 MHz	AMOS 4.3A	AlphaBASIC	317 sec
Alpha Micro	AM-100	WD16	16	2 MHz	AMOS 4.3A	AlphaBASIC	573 sec
Technico	SS-16	9900	16	3 MHz	DOS	Super BASIC 3.0	585 sec
Ohio Scientific	C4-P	6502	8	2 MHz	OS65D 3.2	Level I BASIC	680 sec
Radio Shack	TRS-80 Model II	Z80	8	4 MHz	TRSDOS 1.1.2	Level III BASIC	955 sec
Apple	II PLUS	6502	8	2 MHz	DOS 3.2	Applesoft II BASIC	960 sec
Rexon	RX30	8086	16	5 MHz	RECAP	Business BASIC	1020 sec
Ohio Scientific	C3-C	6502	8	1 MHz	OS65D	Level I BASIC	1346 sec
ISC	Compucolor 8051	8080	8	n/a	DOS	BASIC 8001	1375 sec
Hewlett-Packard	HP-85	Prop	8	n/a	n/a	BASIC	1380 sec
Basic/Four	600	8080	8	n/a	n/a	BASIC	1404 sec
Micro V	Microstar I	8085	8	3 MHz	StarDOS	StarDOS BASIC	1438 sec
Zilog	MCZ-1/70	Z80	8	4 MHz	RIO	Zilog BASIC	1864 sec
Radio Shack	TRS-80 Model I	Z80	8	2 MHz	TRSDOS	Level II BASIC	1928 sec

puter sciences school to the most-widely produced computer in the history of the world: the Radio Shack TRS-80. We have included all of the configuration information we could pin down, as there can be significant differences between different versions of an operating system, for example. The chart only shows the results of testing we have performed personally. Others have run the benchmark on a variety of other equipment, but since we weren't there to witness the test conditions, we opted to leave the data out of the table, interesting as it was.

Try it yourself, and share the results with us. Use any tricks you know to speed up the program execution (multiple statement lines, integer variables, etc.), but don't "improve" on the basic algorithm itself. A canny programmer at an Ohio Scientific store knocked 10% off the execution time by substituting variables for line numbers in this manner:

```
125 LET X1 = 0
126 LET X2 = 230
•
•
•
180 IF L = X1 THEN X2
```

That trick doesn't strike us as being a normal programming technique, so we didn't allow it in our chart, even though it rates an A+ for comprehension of his BASIC interpreter's inner workings.

Include the actual listing of the program, along with all pertinent information regarding operating system version, brand of language, etc. If enough readers send in results, we will tabulate them for publication in a future issue. We would particularly like to share with you data on any runs on large mainframe computer systems. (If it's a time-sharing machine, make several runs during off-peak times and keep the best one.) It would also be interesting to see how different operating systems and/or languages compare, using the same hardware.

The Association of Computer Users (P.O. Box 8003, Boulder, CO 80301) is a non-profit organization that carries benchmarking to a high art. For $150 per year, you can subscribe to *Benchmark Report,* which shows the results of some in-depth testing on a variety of business computer systems. Their benchmarks are broken down into speed tests (CPU and I/O intensive), real life problems (scientific/engineering and accounts receivable) and ease of use (number of keystrokes required for text editing). They seem to have covered all of the important areas except, perhaps, prime number crunching.☐ —TF

Program follows

LISTING 1—Prime Number Cruncher

If the terminal doesn't supply an automatic carriage return/line feed at the right-hand edge, line 230 will cause the display to "stick" after printing only the first few primes.

```
100 REM     INTERFACE AGE's benchmark program to
110 REM     'discover' the first 1000 prime numbers
120 REM
130 PRINT "Starting:"
140 FOR N = 1 TO 1000
150  FOR K = 2 TO 500
160   LET M = N/K
170   LET L = INT(M)
180   IF L = 0 THEN 230
190   IF L = 1 THEN 220
200   IF M > L THEN 220
210   IF M = L THEN 240
220  NEXT K
230  PRINT N;
240 NEXT N
250 PRINT CHR$(7)
260 PRINT "Finished."
270 END
```

A BREAK Service Routine for a KIM-1 with a Teletype

By Ken Wetzel

Frequently a program requires debugging before it operates as intended. The use of software interrupts, or breakpoints, are recognized debugging aids; to this end, many microcomputer monitors include breakpoint routines. The KIM-1 monitor lacks this capability. The program presented here is for use with a KIM-1 and a TTY terminal. It allows the listing of the contents of the internal registers of the 6502 microprocessor unit (MPU) whenever the MPU encounters a breakpoint in the execution of a program. This break service routine returns control to the KIM monitor to permit user intervention before continuing with program execution.

In using a break service routine it is necessary to replace an opcode within the user's program with the breakpoint. Specifically, for the 6502 the opcode for break (BRK) is '00'. Replacing an opcode with the opcode for break is often called 'patching' a breakpoint. Monitors on some other microcomputers have break service routines that automatically patch the breakpoint while saving the original opcode, and then automatically restore the original opcode during execution. Convenient as they are, such routines are limited in the number of breakpoints they can support.

The program I have written requires patching breakpoints into a program and restoring the original opcodes, both manually. Although I was primarily concerned with having a short enough routine to fit in the available RAM that begins at address 1780, the routine also has the advantage of serving a limitless number of breakpoints. What I have attempted to optimize, however, is the format for printing the MPU contents on the TTY. The format is shown in figure 1.

```
02C5 X:39 Y:02 A:61 S:FF N:1 V:0  :1 B:1 D:1 I:0 Z:1 C:1
```

Figure 1. Break Service print out format.

The break service routine has the TTY do a carriage return and a line feed, print the address of the breakpoint encountered, and then print the contents of the X and Y index registers, the accumulator, the stack pointer, and each flag with its status. The undefined 6502 flag is identified with a blank. As soon as the status of the carry flag is printed, control of the KIM-1 is returned to the user via the usual command keys. Examples of this are included in this article.

The use of this break service routine is straightforward. If a program is demonstrating questionable responses, merely change the opcode at the suspect location to 00. When the program is run again, the breakpoint will activate the TTY listing of the MPU registers, provided that execution reaches the breakpoint. I find that a liberal number of breakpoints is desirable when the errant program demonstrates confusing execution.

Since the breakpoints are inserted and removed manually, it is quite helpful to patch a breakpoint over an NOP, even though a BRK can be patched over any opcode in RAM. With this method it is not necessary to repeatedly patch a breakpoint and restore the original opcode to repeatedly execute the program. The debugging ease which results from patching over an NOP makes it very desirable to include them in strategic locations during the initial stages of program development.

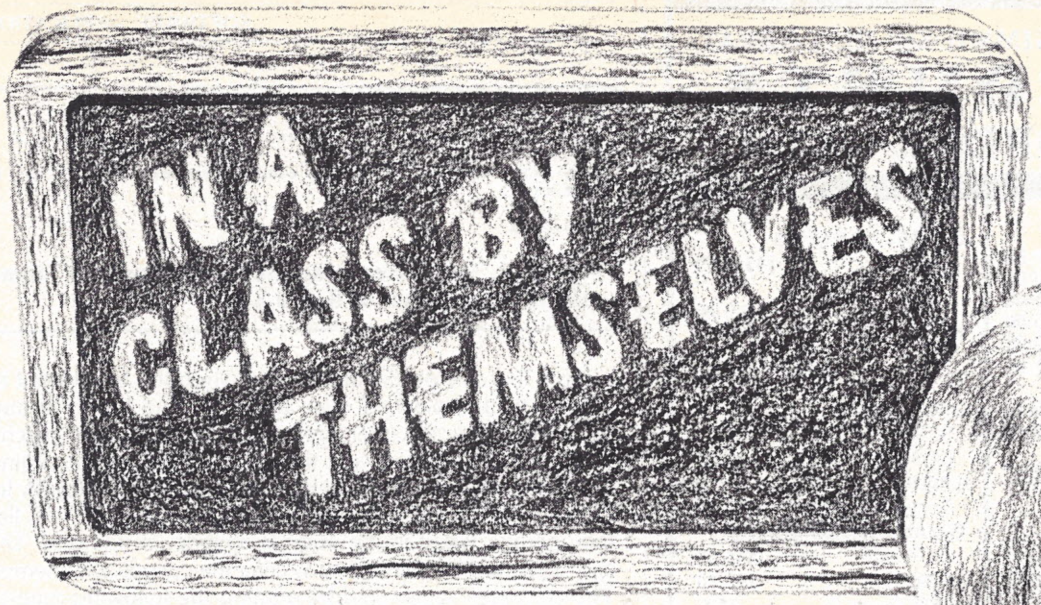
The following example will help to illustrate the breakpoint service operation. Figures 2 and 3 show a sample program before and after patching two breakpoints. One replaced an NOP, and the other a BNE. Figure 4 shows the computer response and operator commands when running the program of figure 3. The first line shows the starting address of the program and the operator's 'G' command to resume program execution. After the computer finished the second line, the operator advanced to the opcode following the BRK by using the return key. The computer then printed the address and opcode on the third line and the operator restarted program execution with the G key.

```
0000  18          TEST   CLC               Nonsense program to
0001  F8                 SED               demonstrate Break Service
0002  A9 39              LDA #  39
0004  AA          REPEAT TAX
0005  69 63              ADC #  63
0007  A8                 TAY
0008  49 63              EOR #  63
000A  EA                 NOP               Space to patch a breakpoint
000B  24 13              BIT z  DATA
000D  D0 F5              BNE    REPEAT
000F  D8                 CLD               Enable KIM TTY routines
0010  4C 64 1C           JMP    CLEAR      Return to monitor
0013  9E          DATA
```

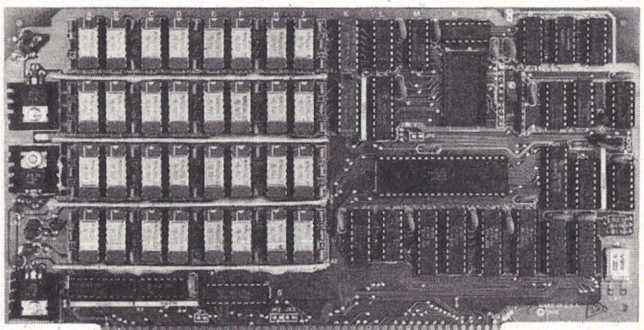
Figure 2. Sample program before patching break points.

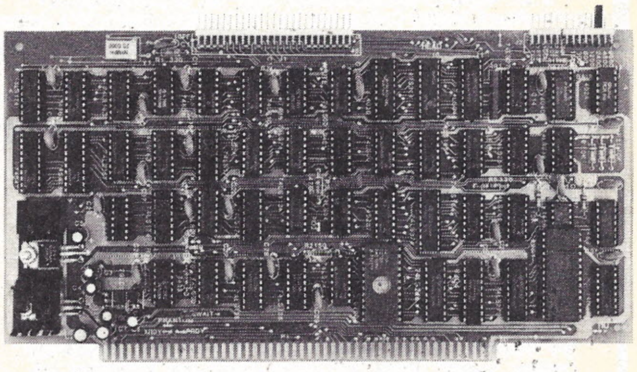
```
0000  18          (TEST) CLC               Nonsense program with
0001  F8                 SED               breakpoints patched in
0002  A9 39              LDA #  39
0004  AA          REPEAT TAX
0005  69 63              ADC #  63
0007  A8                 TAY
0008  49 63              EOR #  63
000A  00                 BRK               Breakpoint 000A
000B  24 13              BIT z  DATA
000D  00 F5              BRK    *          Breakpoint 000D
000F  D8                 *
0010  4C 64 1C           *
0013  9E          DATA

* Ambiguious until BRK at address 000D
is replaced with original opcode.
```

Figure 3. Sample program with break points.



Introducing a whole new school of thought are the new twins of the microcomputer industry . . . the Model Z80/64 CPU and memory board and the Model TWO-X single/double density disk controller. Like degrees and diplomas this new class is an achievement beyond compare.





The Z80/64, a Z80 processor, runs at a full 4 Mz and is supplemented with 64K bytes of dynamic RAM with provisions for vectored interrupts and 2K EPROM, all on a quality S-100 bus board. The Z80/64 has low power requirements and full transparent refresh during unutilized processor time. With CPU and memory on the same board, timing problems and bus noise are eliminated. Who said you can't teach an old system new tricks?

The TWO-X, double/single density disk drive controller, handles four 8″ or three 5″ single or double headed disk drives, and operates single/double density automatically. Transfering files between densities, operating at 2 or 4 Mz, the TWO-X includes a RS232 Serial port. There's a hardware 2708 PROM on board and the software is all ready to go, configured to run without a front panel. It's a real apple polisher.

To send your system to the head of it's class, put these boards in our new "Cool" mainframe, the Model 2018 desk or rack mount cabinets . . . they're in a class by themselves.

So if you are ready to stand above the crowd, call the "Coolest" distributors in the country, CMC MARKETING, or see them at your local retail computer store.

CMC Marketing

10611 Harwin,Suite 406, Houston, Texas 77036 • (713) 995-4960 • TWX: 910-881-7155 TELEX: 762072

```
0000 18 G
000A X:39 Y:02 A:61 S:FF N:0 V:1  :1 B:1 D:1 I:0 Z:0 C:1
000B 24 G
000D X:39 Y:02 A:61 S:FF N:0 V:0  :1 B:1 D:1 I:0 Z:1 C:1 DO.
000E F5

000D DO G
```

Figure 4. Computer response and operator commands for program in figure 3.

Note that this procedure is used when the breakpoint being serviced is patched over an NOP. The fourth through sixth lines show the operation when a breakpoint is patched over an actual opcode and the operator desires to continue execution after the break service is finished. The fourth line shows that after the computer printed the status of the C flag, the operator typed in the original opcode, in this case 'D0', and depressed the period key to enter the opcode into memory. The computer responded by printing the following memory location with its contents, in this case '000E F5'. Now it is necessary to backup to the desired opcode by using the LINE FEED key on the TTY. The last line of figure 4 shows the address of the replaced opcode, the replaced opcode, and the 'G' the operator typed to resume program execution.

```
1780  D8        BRKSRV CLD              Enable TTY monitor routines*
1781  85 F3     SAVE   STA z  F3        Save MPU registers
1783  68               PLA
1784  85 F1            STA z  F1
1786  68               PLA
1787  38               SEC
1788  E9 02            SBC #  02        Correct Program Counter address
178A  85 EF            STA z  EF
178C  85 FA            STA z  FA
178E  68               PLA
178F  E9 00            SBC #  00
1791  85 F0            STA z  F0
1793  85 FB            STA z  FB
1795  84 F4            STY z  F4
1797  86 F5            STX z  F5
1799  BA               TSX
179A  86 F2            STX z  F2
179C  20 2F 1E  PPC    JSR    CRLF      Car. ret. & line feed (KIM subr)
179F  20 1E 1E         JSR    PRTPNT    Print contents of FA & FB (KIM)
17A2  A2 0B            LDX #  0B        Point to end of lookup table
17A4  20 CA 17  PREG   JSR    PSUB      Print lookup table & ':'
17A7  B5 EA            LDA zX EA        Get saved register
17A9  20 3B 1E         JSR    PRTBYT    Print A as 2 hex characters (KIM)
17AC  CA               DEX              Move pointer
17AD  E0 07            CPX #  07        Finished with registers?
17AF  D0 F3            BNE    PREG      No: do another 'PREG'
17B1  A5 F1            LDA z  F1        Yes: get status flags and
17B3  85 F7            STA z  F7        save in temp
17B5  20 CA 17  PFLAG  JSR    PSUB      Print lookup table & ':'
17B8  A9 30            LDA #  30        ASCII zero
17BA  06 F7            ASL z  F7        Flag to ASCII char conversion
17BC  69 00            ADC #  00
17BE  20 A0 1E         JSR    OUTCH     Print A as ASCII character (KIM)
17C1  CA               DEX
17C2  10 F1            BPL    PFLAG     Finished? No: next flag
17C4  20 9E 1E         JSR    OUTSP     Yes: print a space (KIM)
17C7  4C 64 1C         JMP    CLEAR     and return control to KIM
17CA  20 9E 1E  PSUB   JSR    OUTSP     Print a space (KIM)
17CD  BD D9 17         LDA @X TABLE     Get ASCII char from table
17D0  20 A0 1E         JSR    OUTCH     Print a character (KIM)
17D3  A9 3A            LDA #  3A        ASCII colon
17D5  20 A0 1E         JSR    OUTCH     Print a colon (KIM)
17D8  60               RTS

 17D9  43        TABLE                   ASCII  C
 17DA  5A                                  "    Z
 17DB  49                                  "    I
 17DC  44                                  "    D
 17DD  42                                  "    B
 17DE  20                                  "    SPACE
 17DF  56                                  "    V
 17E0  4E                                  "    N
 17E1  53                                  "    S
 17E2  41                                  "    A
 17E3  59                                  "    Y
 17E4  58                                  "    X

* The KIM monitor routines for the TTY don't all execute correctly
if the 'D' flag is set.
```

Figure 5. Break Service Routine

In order to use this break service routine, it is necessary to store the starting address, 1780, in the IRQ vector location 17FE and 17FF of the KIM RAM. However, if an interrupt system using the IRQ is being operated, it is necessary to store the interrupt service routine starting address in the IRQ vector. Then include in the interrupt service routine a test of the B flag and a jump to 1780 when it's set. An example for implementing this appears in the 6502 Programming Manual.

The documentation for this program appears in figure 5. The save registers part of the program is essentially the same as the SAVE routine at location 1C00 in the KIM monitor. The one difference is that the address stored into locations 00FA and 00FB is adjusted by subtracting 2 from the numbers stored when the Program Counter is pushed on the stack. That is necessary to make those registers point to the address of the breakpoint being serviced. (It seems the 6502 adds 2 to the Program Counter before it realizes that the BRK command is actually an interrupt.)

The break service program also uses an unconventional return to the KIM monitor. The usual return address of 1C4F causes the TTY to print the 'KIM' and the address of the breakpoint with '00' on two extra lines. I considered this an unnecessary waste of paper and time. After a bit of trial and error, I found that a jump to address 1C64 eliminates this difficulty.

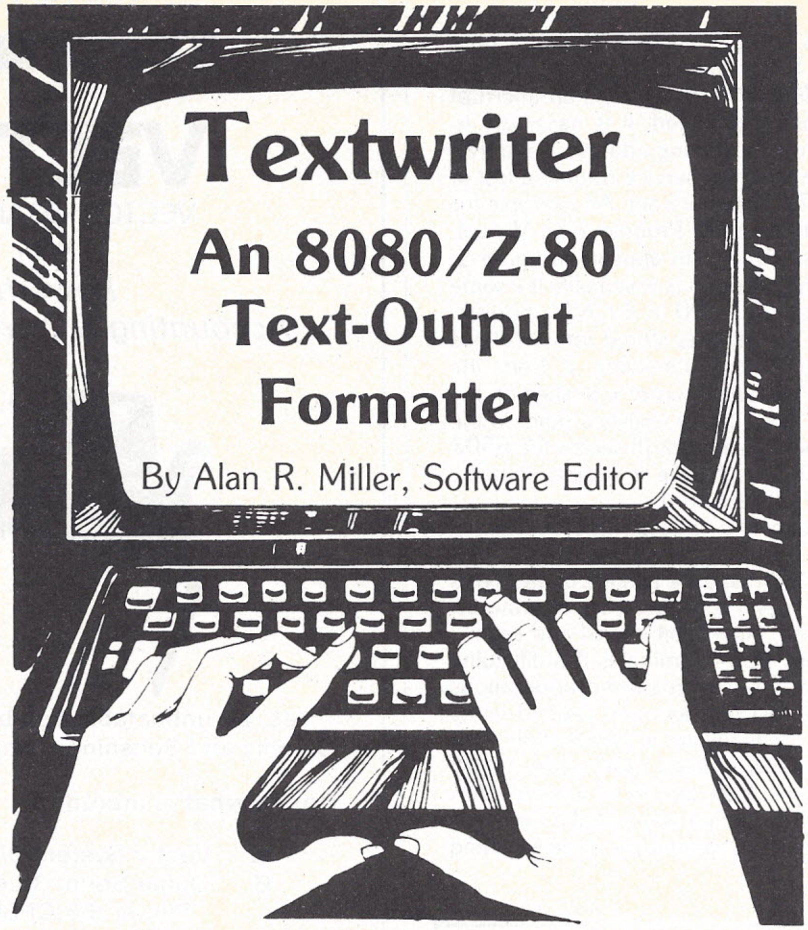
In using the program I have found two distinct applications for it. The first is its use to test registers and flags to verify proper program operation or to locate a malfunction. The second use is the one I actually employ most often: I keep the service routine at location 1780, and fill all unused RAM with BRKs (opcode 00). Then almost any mistake I make is immediately caught. Destroyed programs are virtually a thing of the past.

```
0000  A2 7F     BS BF  LDX #  7F       Set pointer to END
0002  B5 10     MOVE   LDA zX 10       Get byte to be moved
0004  9D 80 17         STA @X BRKSRV   Store in 1780-17FF RAM
0007  CA               DEX             Decrement pointer
0008  10 F8            BPL    MOVE     Finished? No: do another 'MOVE'
000A  A9 00     BRKFIL LDA #  00       Yes: clear A
000C  EA               NOP
000D  4C E5 17         JMP    CLRRAM   Jump to the moved BRKFIL program
0010  D8        BRKSRV                 First line of BRKSRV
 :::
0074  58                               Last line of BRKSRV
0075  8D 00 16  CLRRAM STA @  AL,AH    Store 00 in RAM address AL, AH
0078  EE E6 17         INC @  AL       Next RAM address
007B  D0 F8            BNE    CLRRAM   This page full? No: do 'CLRRAM'
007D  CE E7 17         DEC @  AH       Yes: next page
0080  10 F3            BPL    CLRRAM   Last page? No: do 'CLRRAM'
0082  4C 22 1C         JMP    RST      Yes: jump to monitor
0085  00                               Bytes to be moved to KIM RAM:
0086  00
0087  01                               for the tape interface
0088  00
0089  00
008A  00                               for the NMI vector
008B  1C
008C  4F                               for the RST vector
008D  1C
008E  80                               for the IRQ vector
008F  17        END
```

Figure 6. Program to load break services and fill RAM with BRKs.

The program of figure 6 is the one I use to load my break service routine and to fill RAM with BRKs. It's very easy to use since the tape load routine returns with 0000, the starting address of this program. Just load the program from tape, depress the G key, and then the RUB OUT key. The RUB OUT key is necessary since this program destroys the information stored during initialization of the TTY, and therefore must be re-initialized.

The KIM-1 break service provides a useful routine that fits in a limited amount of memory.□



INTRODUCTION

The availability of modern computers has changed our lives in several ways. One of these changes affects the way we prepare written documents. A business letter may be dictated to a secretary who will type the letter in its final form. Reports may be handled differently because they are longer. In this case, the author will prepare an outline followed by a handwritten rough draft. The next version will then be typed from the rough draft.

If major changes are necessary, the report can be cut apart and taped back together in its new form. A boiler-plate section, describing the capabilities of the company, may be appended to the end of the report. If there are many changes, then the entire report may have to be retyped.

COMPUTERIZED TEXT FORMATTING

Report preparation is considerably simplified if a text-formatting computer program is available. With this approach, the author's handwritten draft is typed into a computer under control of the system editor. Format commands are embedded in the text at this time. A separate, text-formatting program is then invoked to generate the finished document. This final result appears at the line printer or is stored on disk as a separate file.

Major rearrangement of the text is easily accomplished if the system editor has a block-move command. The report never has to be retyped.

One text formatter, the Electric Pencil, was reviewed in the August 1978 issue of INTERFACE AGE. This program combines an editor of sorts with a text formatter. It cannot, however, be used with a serial video console since it requires a memory-mapped video screen.

There are several text formatters available for the CP/M operating system. One of these is Word Star by Micropro. Another formatter, Tex (reviewed in the May 1979 issue of INTERFACE AGE) is provided by Digital Research. A similar text formatter program, Textwriter, is available from Organic Software of Livermore, CA. The same program is available from Micropro under the name of Tex-Writer. There are separate versions available to use on CP/M, Micropolis and North Star operating systems. A TRS-80 CP/M version is also provided. At least 32K bytes of memory should be available.

Textwriter is similar to Tex. Both can convert an existing work file that is stored on disk into a finished file. The work file is previously prepared with the system editor. Textwriter can operate on either of two file formats. In the standard format, each line of the work file is terminated with a carriage-return, line-feed combination. But Textwriter can also format files that were prepared with Electric Pencil. The finished file can be printed during the formatting process or it can be saved as a separate disk file to be printed at a later time.

THE WORK FILE

The work file is created from the rough-draft manuscript by using the system editor. The default file-name extension is TEX for the Digital Research Tex formatter. But since there is no default extension name for Textwriter, one might choose the name TXT. This will be an easy way to distinguish Tex work files from Textwriter work files. This distinction is necessary since the two programs use different formatting commands.

The text is entered directly into the work file without regard to form. If a word is too long to fit at the end of a line, it is placed on the next line. It is not split with a hyphen.

Textwriter accepts about 50 formatting commands. Some of these are automatically set to their default values, but they may be redefined within the text. The general commands for text formatters typically begin with a decimal point. Textwriter is different in that the commands begin with an excla-

mation point. If you don't like the exclamation point, however, you can change it to something else with the COMCHAR command. The statement:

!COMCHAR “ ”

will change the command character from the exclamation point to a decimal point.

There will generally be a block of formatting commands at the top of the work file that are used to set the desired values. The manuscript for this present article was formatted with Textwriter. The following commands were given at the beginning of the work file:

```
!margins 10 64          <margins at columns 10 and 64>
!head 8                 <8 lines at top of page>
!foot 5                 <5 lines at bottom of page>
!ignore                 <don't print next line>
Feb 11,80               <version>
!tty                    <fake form feed with line feeds>
!justify                <align right column>
!pagenum 1 3 "B"        <number pages at bottom>
!skip 5                 <skip 5 lines>
!center
Textwriter
An 8080/Z-80 Text-Output Formatter
Review by Alan R. Miller
Software Editor
!spacing 2              <double space>
!skip 3
INTRODUCTION
!par 3 0                <paragraph, indent 3>
```

Additional formatting commands are included within the text. Some of the more useful commands are:

!par	Start a new paragraph
!literal	Use next section as is
!pageifnot 6	Start a new page if 6 or less lines are left
!skip 2	Skip 2 lines
!spacing 1	Single space the text
!spacing 2	Double space the text
!left 5	Indent left margin 5 spaces
!right 5	Indent right margin 5 spaces
!resetmar	Reset margin to regular width
!footnote	Make a footnote of next section
!* <comment>	A one-line comment

LONG REPORTS

ASCII files that are larger than about 30K bytes are cumbersome to deal with. This is especially true if the system editor is not disk oriented, because, in this case, the entire file must be loaded into memory at one time. Editors like ED, Word-Master, EDIT-80, or ED-80 are disk oriented. Consequently, a small portion of the file can be copied from disk into memory for editing. Even with this type of editor, it is much more convenient to edit a file if it can be entirely loaded into memory.

The work file for a long report or even one chapter of a book can easily exceed 30K bytes. This size is too large to fit into memory. The solution, in this case, is simple if Textwriter is available. The work file is generated as a sequence of conveniently sized disk files. The last line of each file contains a CHAIN command which gives the name of the next file in the series. For example, suppose that chapter 4 of a book were broken up into two parts called:

CHAPT4A.TXT and
CHAPT4B.TXT

The last line of the file CHAPT4A.TXT would be:

!CHAIN “CHAPT4B.TXT”

Chapter 4 can be formatted into a single disk file with the system command:

A>TW LST:=CHAP4A.TXT

The finished file will appear at the line printer (LST:) and will include both parts of chapter 4. When the formatter reaches the end of the file CHAPT4A.TXT, the chain command will direct it to the file CHAPT4B.TXT. The combined finish file can alternately be formed into a disk file called CHAP4.PRN by giving the command:

A>TW CHAP4.PRN=CHAP4A.TXT

Notice that both the destination filename and the source filename must be given in the command line. Furthermore, both filename extensions must be given.

EMBEDDED COMMANDS

Most of the Textwriter commands begin with an exclamation point and appear on separate lines from the text. In these cases there is little chance that commands will be confused with text. There are, however, three Textwriter commands that could cause problems. These commands do not utilize the exclamation point. Furthermore, they are embedded within the regular text rather than being on a separate line.

One of the embedded commands is used to underline a portion of text. The particular section of the work file is enclosed with a pair of braces. The finished document will contain the desired underlining, not the original braces. If brace symbols are desired in the final text, then the underline symbols will have to be changed. This is performed with the UNDERLINE command. After the command:

!UNDERLINE “[]”

appears, then the bracket pairs will define the text to be underlined instead of the braces. More importantly, the braces can now appear in the final text.

A similar situation occurs with the tilde and the at-sign. The tilde is used to indicate a space between two words that must appear on the same line. Textwriter might break up the section:

February 28, 1980

so that the month and day appear at the end of one line and the year at the beginning of the next. Using a tilde instead of the spaces will guarantee that the date will be printed entirely on the same line:

February 28, 1980

The at-sign is used for non-standard tab stops. Both the tilde and the at-sign cannot appear in the final text unless the corresponding commands are redefined. For example, the statements:

!SPACEBAR “\”
!TABCHAR “&”

will change the space character to a backslash and the tab character to an ampersand.

A powerful feature of Textwriter is the ability to input data from the console or from a separate disk file during the formatting process. With this feature, a standard business letter

can be run many times. Each time the letter is printed, the name and address of a different recipient can be read from the system console or from a separate disk file.

TABLE OF CONTENTS

Long reports and books need a table of contents. The Textwriter command CONTENTS is used for this purpose. It operates in a curious way, however. The title for each section or subsection must be entered twice, once for the actual entry, and once for the table entry. For example, the command pair might look like this:

```
!SKIP
!CONTENTS  2  "3.4 Passing Data on the Stack"
3.4 Passing Data on the Stack
!PAR
• • •
```

The number 2 in the CONTENTS line causes the subheading to be indented two spaces.

The necessary duplication of the heading can be easily accomplished with the Q-buffer command of the Word-Master editor. At the end of the report, the CONTENTS command is given without arguments to force printing of the actual table of contents. The corresponding chapter numbers, if any, and the page numbers are printed opposite the given headings. A row of dots connects the heading to the page number.

GENERATION OF AN INDEX

The preparation of an index for a book or a long report can be tedious. The global-search command of the system editor can be used on the finished file, if it was saved on disk during the formatting step. But a better way is to use the INDEX command of Textwriter. Each index entry or subentry is defined with a command such as:

```
!INDEX  "stack"
```

An index entry can be followed by a comma and a subentry:

```
INDEX  "flag, carry"
INDEX  "flag, zero"
INDEX  "flag, parity"
```

The main entry is only given once in the resulting index:

```
flag,
  carry, 2-3
  parity, 2-5
  zero, 2-4, 4-5
```

Then, a final INDEX command near the end of the document will product the complete index.

CONCLUSION

This reviewer has just finished a book on assembly language programming. The entire manuscript was written and edited with Word-Master and formatted with Textwriter. The task was made easier by some of the unique features of Textwriter. In particular, the book's author and title were printed on the top line of each page. The chapter number or appendix letter and the page number were printed at the bottom of each page. The CHAIN command was used to combine the work files of longer chapters into single finished chapters.

One task has not yet been completed. This is the preparation of an index. I expect that the INDEX command of Textwriter will greatly simplify this step.☐

THE DUST WRITER

Continued from Page 68

PROGRAM LISTING

```
  ?EDI
START INPUT
*L
FILE NAME=EC2
*W
        ORG 8000H
        ORR 2C00H
CIN     EQU 2010H
DOS     EQU 2028H
        LXI SP,9000H
        XRA A
        STA POS
TEST:   CALL CIN
        CPI 3
        JZ DOS
        CALL PRINT
        JMP TEST
BASIC:  MOV A,E
TABLE   EQU 8200H
LENGTH  SET 208
PRINT:  PUSH H
        PUSH D
        PUSH B
        LXI H,TABLE
        ANI 7FH
        CPI 0DH
        JZ RETURN
        CPI 20H
        JC EXIT
        JZ SPACE
        MVI C,LENGTH
TLOOP:  CMP M
        INX H
        JZ MATCH
        DCR C
        JNZ TLOOP
        JMP EXIT
MATCH:  LDA CURNT
        MOV B,A
        CALL LIFT
        CALL INDEX
PLOOP:  MOV A,M
        ANI 40H
        CNZ LIFT
        MOV C,M
        CALL MOVE
        CALL DROP
        INX H
        MOV A,M
        RAL
        JC PLOOP
        MOV A,B
        STA CURNT
EXIT:   POP B
        POP D
        POP H
        RET
;
SPACE:  LXI H,SPCS
        INR M
        JMP EXIT
;
INDEX:  XCHG
        LXI H,SPCS
        MOV C,M
        MVI M,0
        INX H
        MOV A,M
        ADD C
        INR A
        MOV M,A
        XCHG
        MVI E,203Q
        MOV A,C
        CPI 1
        CNC HPOS
        MOV A,B
        ANI 7Q
        ORI 350Q
        MOV C,A
        MOV A,M
        ANI 100Q
        JNZ INSKP
        MOV A,C
        ANI 370Q
        MOV C,A
INSKP:  CALL MOVE
        MOV A,B
        ANI 207Q
        MOV B,A
        MOV A,M
        ANI 40H
        CZ DROP
        RET
;
LIFT:   MVI A,80H
        OUT 24
        ORA B
        MOV B,A
        MVI A,20
        CALL DELAY
        RET
;
```

```
DELAY:   PUSH B
DLOOP1:  MVI C,80
DLOOP2:  DCR C
         JNZ DLOOP2
         DCR A
         JNZ DLOOP1
         POP B
         RET
;
MOVE:    MOV A,B
         SUB C
         ANI 77Q
         RZ
         CALL MOVEX
         CALL MOVEY
         CALL MOVEX
         CALL MOVEY
         CALL MOVEX
         CALL MOVEY
         CALL CNTY
         CALL MOVEX
         CALL CNTX
         JMP MOVE
;
DROP:    XRA A
         OUT 24
         ORA B
         RP
         MOV B,A
         MVI A,0
         CALL DELAY
         RET
;
MOVEX:   MOV A,C
         RAL
         ANI 80H
         MOV E,A
         MOV A,B
         ANI 70Q
         MOV D,A
         MOV A,C
         ANI 70Q
         SUB D
         RZ
         JC REVX
         INR E
         INR E
REVX:    INR E
         MVI D,2
MXLOP:   MOV A,E
         OUT 24
         MVI A,20
         CALL DELAY
         DCR D
         JNZ MXLOP
         RET
;
MOVEY:   MOV A,B
         ANI 7
         MOV D,A
         MOV A,C
         ANI 7
         SUB D
         RZ
         MOV A,E
```

```
         JC REVY
         ADI 8
REVY:    ADI 4
         ANI 374Q
         MOV E,A
         OUT 24
         MVI A,14
         CALL DELAY
         RET
;
CNTX:    MOV A,E
         ANI 3
         RZ
         DCR A
         MVI A,-8
         JZ CNSKP
         MVI A,8
CNSKP:   ADD B
         MOV B,A
         RET
;
CNTY:    MOV A,E
         ANI 12
         RZ
         SUI 4
         MVI A,-1
         JZ CNYSKP
         MVI A,1
CNYSKP:  ADD B
         MOV B,A
         RET
;
RETURN:  LDA POS
         CPI 1
         JC EXIT
         MVI E,201Q
         CALL HPOS
         OUT 24
         STA POS
         JMP EXIT
;
HPOS:    MOV D,A
HLOOP1:  MVI C,40
HLOOP2:  MOV A,E
         OUT 24
         MVI A,18
         CALL DELAY
         DCR C
         JNZ HLOOP2
         DCR D
         JNZ HLOOP1
         RET
;
SPCS:    DB 0
POS:     DB 0
CURNT:   DB 0
         END
*E
  ?EDT
START INPUT
*L
FILE NAME=TBL
*W
         ORG 8200H
         ORR 2E00H
```

```
TABLE:   DB "@",324Q,222Q,232Q
         DB 233Q,243Q,245Q,236Q
         DB 216Q,205Q,201Q,210Q
         DB 240Q
         DB "A",204Q,226Q
         DB 244Q,240Q,302Q,242Q
         DB "B",206Q,236Q,245Q
         DB 243Q,203Q,243Q,241Q
         DB 230Q,200Q
         DB "C",345Q,236Q,216Q
         DB 205Q,201Q,210Q,230Q
         DB 241Q
         DB "D",206Q,236Q,245Q
         DB 241Q,230Q,200Q
         DB "E",340Q,200Q,206Q
         DB 246Q,333Q,203Q
         DB "F",206Q,246Q,333Q
         DB 203Q
         DB "G",332Q,242Q,240Q
         DB 210Q,201Q,205Q,216Q
         DB 246Q
         DB "H",206Q,346Q,240Q
         DB 303Q,243Q
         DB "I",310Q,230Q,320Q
         DB 226Q,316Q,236Q
         DB "J",301Q,210Q,230Q
         DB 241Q,246Q
         DB "K",206Q,346Q,213Q
         DB 240Q
         DB "L",340Q,200Q,206Q
         DB "M",206Q,224Q,223Q
         DB 224Q,246Q,240Q
         DB "N",206Q,205Q,241Q
         DB 240Q,246Q
         DB "O",301Q,205Q,216Q
         DB 236Q,245Q,241Q,230Q
         DB 210Q,201Q
         DB "P",206Q,236Q,245Q
         DB 244Q,233Q,203Q
         DB "Q",301Q,205Q,216Q
         DB 236Q,245Q,242Q,220Q
         DB 210Q,201Q,322Q,240Q
         DB "R",206Q,236Q,245Q
         DB 244Q,233Q,203Q,213Q
         DB 240Q
         DB "S",301Q,210Q,230Q
         DB 241Q,242Q,233Q,213Q
         DB 204Q,205Q,216Q,236Q
         DB 245Q
         DB "T",320Q,226Q,306Q
         DB 246Q
         DB "U",306Q,201Q,210Q
         DB 230Q,241Q,246Q
         DB "V",306Q,202Q,220Q
         DB 242Q,246Q
         DB "W",306Q,200Q,222Q
         DB 223Q,222Q,240Q,246Q
         DB "X",201Q,245Q,246Q
         DB 306Q,205Q,241Q,240Q
         DB "Y",320Q,223Q,205Q
         DB 206Q,346Q,245Q,223Q
         DB "Z",306Q,246Q,245Q
         DB 201Q,200Q,240Q
         DB 0
         END
```

MICRO-MARKET ADS SELL YOUR PRODUCTS

A new format has been established for the Micro-Market section. All ads are now 2 inches wide by 3 inches deep. Price is $200. Submit ads with check or money order to:

INTERFACE AGE Magazine
Micro-Market Ads
P.O. Box 1234
Cerritos, CA 90701

HOW CAN I GROW B TREES?

ORDER MICRO B +™

• Get fast insertion, retrieval and deletion of index entries.

• Never need to reorganize your index; no matter how often it is updated.

MICRO B+ SOURCE CODE $195
MICRO B+ DEMO DISK $25
SHIPPING $2 USA. $5 FOREIGN

Available in 8" format for **CBASIC-II** and **MICROSOFT** Basic Version 5.

FAIR COM 2606 Johnson Drive
Columbia, Mo. 65201
314-445-3304

Check/VISA/Mastercharge

Moonshadow Text Formatter for UCSD Pascal™ Systems

The UCSD Pascal™ system contains a screen-oriented text editor which is convenient, but which is not suited to word processing. It cannot underline, paginate automatically, or perform other essential text-processing functions. The **Moonshadow Text Formatter** (MTF) from Merrimack Systems solves this problem.

With the Moonshadow Text Formatter, documents produced with the screen editor are *post-processed* to provide these missing functions. It takes standard Pascal text files, operates on them, and sends fully formatted text output to the console display, a printer, or a disk file.

Moonshadow Text Formatter provides, in addition to a full range of formatting functions, *advanced features* including the combination of files into one document, variables in text (for form letters), and output character translation.

The Moonshadow Text Formatter is written in UCSD Pascal™, and is available for North Star Apple II, LSI-II, and 8080/Z80 systems with IBM Format 8" floppy disks.

All this for **$125.00** *from*

Merrimack Systems
POB 5218
Redwood City, CA 94063
(415) 365-6281

California residents should include 6% sales tax.

Also available: North Star Pascal personalization for SOL/SOLOS — just transfer three files and you're up with Pascal. $25.00

Power Supplies

5 volts @ 25 amps
SURPLUS UNITS IN LIKE NEW CONDITION

INPUT: 115 vac 43 - 63 hz
OUTPUT: 5 vdc 0.25 amps
REGULATION: 0.1% + 5 mV
NL-FL, ± 0.1% ± 5mV for 10% Input change
RIPPLE: 2mV RMS max., 20 mV P-P max.
STABILITY: Typically 10 mV for eight hour period

REMOTE SENSING, REMOTE VOLTAGE ADJUSTMENT, OVERLOAD PROTECTION and OVERVOLTAGE PROTECTION
ALSO AVAILABLE IN 12 VDC @ 13 AMPS $79.00 per unit

79.⁰⁰ per unit

HOUSE MARKED DARLINGTON TRANSISTOR
SPECIFY NPN or PNP
5 AMPS 65 WATTS
80 VOLTS
SIMILAR TO: TIP 121NPN TIP 126 PNP TO220
75¢ each

22/44 EDGEBOARD CONNECTOR
TIN SOLDERTAIL .156" x .200"
LARGE QUANTITIES AVAILABLE
$1.35 each 10 for $12.50

1½ AMP FULL WAVE BRIDGE RECTIFIER
400 PIV
2 for $1.00

REED RELAYS
9-15 VOLT D.C. LATCH
NORMALLY OPENED
CONTACTS:
1 amp max. switch
2 amp max. carry
BODY SIZE:
approx. 1 inch long
¼ inch high

S.P.S.T. 1500 ohm coil $1.15 each
D.P.S.T. 1200 ohm coil $1.30 each
3.P.S.T. 750 ohm coil $1.50 each

ALL ELECTRONICS CORP.
905 S. Vermont Ave. • Los Angeles, Calif. 90006
DEPT. IF-1 (213) 380-8000

STORE & WAREHOUSE HOURS
Mon. - Fri. 9 AM - 5 PM
Saturday 10 AM - 3 PM

TERMS
• Quantities Limited
• Min. Order $10.00
• Add $1.50 Shipping USA
• Calif. Res. Add 6%
• Prompt Shipping

SEND FOR OUR FREE CATALOG

KEYCAPS ENGRAVED

for Cherry and Keytronic keyboards

24 Hour Service

ARKAY ENGRAVERS., INC.
2073 Newbridge Road
Bellmore, New York 11710
[516] 781-9859

Connect your TRS-80, Apple or ANY other computer to the phone lines.

USR-330 Originate—Auto-Answer Modem

- 0-300 Baud
- Stand Alone
- RS232
- 1 Year Warranty
- Crystal Controlled
- Bell 103/113

$339

- State of the Art LSI circuitry
- 5 stage active filters

FCC certified for direct connection to phone lines via standard extension phone jack

Call or write for free literature

U.S. ROBOTICS, INC.
1035 W. LAKE ST.
CHICAGO, ILL. 60607
(312) 733-0497

GET Paid for using your Computer

FUN! Easy

RUSH COUPON FOR FREE FACTS

GREAT SPARE TIME

Send today to — DAR-IA6
3110 Fulton Ave, Sacramento CA 95821

CASH

NAME

STREET

CITY

STATE ZIP

SAVE MORE THAN 20%!

NORTH STAR - INTERTUBE
THINKER TOYS - MICROTEK
The smartest computers at the smartest price
Quad & Double Density

	LIST	ONLY
HORIZON-1-32K-D kit	$1,999	$1585
HORIZON-2-32K-D kit	2399	1905
Assembled & tested	2765	2195
HORIZON-2-32K kit QUAD	2799	2225
Assembled & tested	3215	2555
Pascal for North Star on Disk		49
Powerful North Star BASIC		FREE
TEI PT 212 Computer 5 MHz	8000	6250
Thinker Toys DISCUS 2/D, A&T	1149	949
Discus/2+2 1.2 megabytes, A&T	1549	1299
Measurement System Memory, A&T, 4MHz 64K		640
Godbout Memory		Call for Price
INTERTUBE II Smart Terminal	995	780
MICROTEK Printer	750	675
ANADEX Printer	995	875
Florida Data Printer, 600 cps, 4300		Call for Price
MARYELLEN Word Processor		Your Best Buy 38
Textwriter III		125
EZ-80 Tutorial, Learn Machine Language		25
PDS for North Star, Better than CP/M		99
Compiler for Horizon Secret Superfast Code		100

10• off software prices with computers

VERBATIM the best Diskettes	Box of 10	29
Which Computers are best? BROCHURE		FREE
North Star documentation refundable w/HRZ		20

AMERICAN SQUARE COMPUTERS
Kivett Dr Jamestown, NC 27282 919-883-1105

BUSINESS SOFTWARE

CP/M AND MODEL II COMPATIBLE
NEWLY ENHANCED VERSIONS

MEDICAL MGMT SYSTEM	$495
DENTAL MGMT SYSTEM	$495
REAL ESTATE MULTI-LIST	$495
INSURANCE AGENCY MGMT	$495
LEGAL TIME ACCOUNTING	$495
GENERAL LEDGER	$495
ACCOUNTS PAYABLE	$495
ACCOUNTS RECEIVABLE	$495
PAYROLL PROCESSING	$495
INVENTORY MANAGEMENT	$495
WORD PROCESSING/NAD	$495
DATABASE MANAGEMENT	$495
HIGH-SPEED UTILITY SORT	$300
NEW. LATEST CBASIC-2	$ 99
NEW. LATEST MBASIC 5.1	$300
NEW! MBASIC COMPILER	$350
**CPM 2.0 FOR MODEL II	$170
COMPLETE SOFTWARE CATALOG	5
COMPLETE HARDWARE GUIDE	5

UNIVAIR, INC. 314-426-1099
10327 LAMBERT INTL AIRPORT
ST. LOUIS, MISSOURI 63145 USA
MASTER CHARGE/VISA CARDS O.K.

MONITORS, HIGH resolution B/W-color Sony, Sharp, Hitachi, S-C, all sizes, most from stock. Lowest prices. Catalog $1.00. M-C/Visa. Video Technology, Inc., 14422 N.W. 7 Avenue, Miami, FL 33168, (305) 688-6618.

WANTED: UTILITY billing package for TRS-80 Model II running under CP/M or TRSDOS. Send information and price to: Edwin Phillips, 928 Davis, New Madrid, MO 63869.

TRS-80 QUALITY SOFTWARE. Data base manager/report generator: no user programming, $69, MOD-II $199. A/R, $69, MOD-II $149. Inventory, key random access, $99, MOD-II $149. Word processor, $49. Mailing list, $59, MOD-II $99. Manual $5, MOD-II $10. Micro Architect, 96 Dothan St., Arlington, MA 02174.

VIRTUOSO PET. 8/16/32K Commodores play Bach, Beethoven, etc. Music Box & Allen Animation have 4 programs, $10 each. Order both for free CB2 connector. Add $1 shipping. Allen Computer Products, Box 22836B, Livonia, MI 48151.

WEST GERMAN computer store seeking vendors for hardware and software. Write Kleinofen Elektronik, Koelner Str. 49, D-4000, Duesseldorf 1, West Germany.

16K-RAM COMPUTER — New personal computer. Limited quantity. $300 off factory list. Sale price $399.00. First come basis. Call (312) 382-2192. Write J.H.S., 515 W. Shadylane, Barrington, IL 60010.

TRS-80 SPEED CONTROL plans — very simple circuit allows wide range of control over speed of listing or running programs. Connect to TRS-80 port, or any Z-80. $15.00 check to: Golden Circuits, P.O. Box 404, Golden, CO 80401.

BOWLING SECRETARY for North Star. Most complete package available. Handles prior or self-establishing averages, blinds, vacancies. Can input number of games before handicap becomes official. Keeps 3 places for scratch/handicap high game/series/average for both men and women. Prints weekly league sheet, individual records and sub report. Only $60. JCS Computer Systems, 4815 F.M. 2351, Suite 202, Friendswood, TX 77546.

DOUBLE-SIDED SHUGART 8" floppy drives with single and double density capability. Will store up to 1.2 megabytes per drive. Currently sell for $940 each. Asking $1600 for the pair. New, in cartons. Dennis Dickerson, Rt. 1, One Shannon Valley, Crowley, TX 76036, (817) 737-1313.

TRS-80 PINBALL game, requires Level-II 16K. Actual speed assembler program with flippers, bumpers, rollovers, targets. $19.95. Grania Software, 16, Jackson Court, Kanata, Ontario, Canada K2K 1B7.

TRS-80 EXCHANGE — Used TRS-80s bought and sold. Level I's, II's and peripherals available with warrantees. For a quick response, call Roy (203) 669-0726.

MICRODEX is a classified advertising section available to readers wishing to buy, sell or trade hardware, software, peripherals, accessories, etc.
Price is $1 per word, with a 25 word minimum. The first two words are printed bold. Additional bold type is $5 per word.
Send ad copy with check or money order to INTERFACE AGE Magazine, Dept. M-I, 16704 Marquardt Avenue, Cerritos, CA 90701.

CIRCLE INQUIRY NO. 86

ADVERTISER INDEX

*Manufacturer requests factory-direct inquiry.

Introducing our new Alphanumeric Line Printer: the Anadex Apple/Sider. Specifically designed for the Apple Computer Business System, Apple/Sider provides the 96-characters-per-line required by AL, AP and GL Software of the Apple Controller.

Based on the time-proven Anadex Model DP-8000 printer, this new unit (Model DP-8000-AP) bi-directionally prints the complete 96 ASCII character set in 96 columns at 134 CPS or 84 LPM nominal throughput.

Apple/Sider features a heavy duty printing head (100M character life) that can print the original plus up to 3 copies on paper whose width can range from under 3 inches to 9½ inches. Paper can be loaded either through the rear or through the bottom of the printer.

Standard features include a 1K character buffer storage (optional, an additional 2048 character storage for CRT dump or similar uses), internal programmable top of form and skip over perforation controls, double width printing, and a self test feature which checks all memory and printing functions.

For complete details and the name of your nearest dealer, contact us today. Inquiries from qualified dealers are welcomed.

Anadex®

CIRCLE INQUIRY NO. 4

Apple® Computer compatible Printer:

® Registered Apple Computer, Inc.

anadex apple/sider

ANADEX, INC. • 9825 DeSoto Avenue • Chatsworth, California 91311, U.S.A. • Telephone: (213) 998-8010 • TWX 910-494-2761
ANADEX, LTD. • Dorna House, Guildford Road • West End, Woking, Surrey GU24 9PW, England • Tel: Chobham (09905) 6333 • Telex: 858762 ANADEX G


Make your micro
MIGHTY!
GRAHAM-DORIAN SOFTWARE
REFERENCE MANUAL
Get the most out of your microcomputer with Graham-Dorian Business software.
At any given time, your hardware is only as useful as the software you insert in it.
So it pays to rely on Graham-Dorian, the software that gets your micro performing to its fullest — almost like a mini.
Graham-Dorian, the industry leader, offers highly detailed and well-documented programs. All pretested on the job. Each so comprehensive that it takes little time to learn to run a program — even for someone who's never operated a computer before.
Programs are compatible with most major computers using CP/M disk operating systems, and come in standard 8'' or on various mini-floppy disks. Each package contains the software program in INT and BAS file form plus a user's manual and hard copy source listing. Graham-Dorian stands behind dealers with technical advice.
Yes, there's a world of difference in business software. Graham-Dorian has more per-package capabilities and more packages. (With new ones added every few months.)
The Graham-Dorian line now includes these packages:
Medical
Dental
Surveying
Inventory
Payroll
Apartment Management
Construction Job Costing
Accounts Receivable
Accounts Payable
General Ledger
Cash Register
CBASIC-2
Ask your dealer for a demonstration soon.
GDSS
Graham-Dorian Software Systems, Inc.
211 North Broadway / Wichita, KS 67202 / (316) 265-8633
CIRCLE INQUIRY NO. 32